

SpikeMark™ v2.5

User Manual



This page has intentionally been left blank.

Table of Contents

Welcome to SpikeMark	7
A Word About Safety	7
System Requirements.....	8
Installing SpikeMark (in Windows 7).....	8
Online Tutorials	13
SpikeMark's Newest Features	13
Spikes.....	13
Time-Based Cues	14
Streamlined Interface	14
3-Dimensional Stage View	15
Reset Position.....	16
Target Tolerance Mode.....	16
Stagehand FX Support	16
Quick Start	18
Setting Up Physical Connections.....	18
Setting Up a Network Connection (in Windows 7).....	19
Setting the IP Address of Your Computer	19
Establishing a Connection to a Stagehand Motor or FX.....	23
Confirming Encoder Feedback	25
Setting the Position Scaling Factor	26
Setting Soft Limits	28
Setting a Motor's Default Speed Parameters	29
Writing Two Sample Cues.....	32
Running the Two Sample Cues.....	34
Basic Tuning	36
SpikeMark's Main Screen.....	37
Stage Model Viewer	37
Motor Editing Pane	37
Cue Controller.....	37
Cue Grid.....	37
FX Editing Pane.....	38
Splitter Panes	39
The Stagehand Motor.....	40
Adding a New Stagehand Motor.....	40
Stagehand Motor Configuration	40
Renaming a Motor	42
Motor Status.....	43
Manually Moving a Motor Using the On-Screen Controls	44
Scaling Motor Position	45
Setting Motor Software Limits: The Min Reverse and Max Forward Positions	48
Resetting the Motor's Position Value	50
Motor Tuning.....	51

Proportional Gain.....	52
Derivative Gain.....	52
Derivative Sampling	53
Integral Gain	53
Integral Limit	54
Filter Loaded.....	54
Setting the Max Position Error	57
Setting the Motor Cue Completion Mode.....	58
Target Tolerance	58
Strict Timing.....	58
Setting a Motor's Active Mode.....	60
Changing the Status Refresh Rate.....	62
The Stagehand FX.....	65
Adding a New Stagehand FX	65
Stagehand FX Configuration	67
Renaming an FX	68
Renaming FX Inputs and Outputs.....	68
FX Status.....	70
Manually Activating FX Outputs.....	71
Setting the FX Active Mode	71
FX Input Action	73
Deactivating Stagehand FX Action	73
Removing a Stagehand FX Input Action	74
Soft Stopping FX Action	74
Network Connections.....	77
Editing Stagehand Network Settings	77
Connecting to a Stagehand Motor or FX	79
Disconnecting from a Stagehand Motor or FX.....	80
Writing Cues	82
Adding Cues.....	83
Adding Motor Movements to a Cue	84
Motor Movement Parameters	85
Position	85
Speed.....	86
Acceleration	86
Ramp Time.....	86
Total Time.....	86
Changing Motor Position In a Cue	87
Changing How Fast a Motor Moves Within a Cue.....	88
Specifying the Speed of a Motor Directly	88
Specifying <i>How Long</i> the Motor Should Take to Reach the Target Position.....	89
Changing Cue Time	90
Adding an FX Movement to a Cue	91
Removing a Motor's Movement from a Cue.....	93
Deleting a Cue.....	94
Cue Links	96

Parent Cues	96
Child Cues.....	96
Linking Cues by Completion	97
Linking Cues by Time	99
Linking Cues by Motor Position.....	100
Linking Cues by FX Input	101
Deleting a Cue Link	102
Multi-Speed/Multi-Position Links	103
Changing the Speed of a Motor—Manually (Without a Link).....	103
Changing the Speed of a Motor—Automatically	104
“Ping-Pong” a Motor—Changing a Motor’s Position	105
The Stage Model Viewer	107
Navigating the Stage Model Viewer.....	108
Setting the Stage Dimensions	111
Making a Winch Model	112
Making a Lift Model	115
Making a Curtain Model	118
Making a Turntable Model	121
Spikes.....	125
Adding Spikes within Cues—Using the Spikes Position Button	125
Adding Spikes within Cues—Using the Motor Editing Pane.....	129
Changing a Spike Position.....	133
Selecting Spikes Using the Stage Model Viewer	135
Running Shows	138
Creating a New Show	138
Setting up the Workspace	138
Rearranging the Cue Grid	140
Loading a Cue	141
Visual Indicators of Primed and Ready Movements	143
Running a Cue.....	145
What Happens While a Cue Is Running	146
What Happens as Motor Movements and Cue(s) Complete.....	148
Deciphering Stagehand Motor Status During a Show	151
Restoring a Cue.....	153
Troubleshooting	157
Contact Us.....	157
The Stagehand Network Connection Doesn’t Work	157
A Motor Didn’t Start To Move When I Ran A Cue	158
SpikeMark Showed A Strange Error Message Then I Had To Quit.....	158
I Can’t Find The Log File To Email It To Support.....	159
A Motor Starts To Run In A Cue But Stops Abruptly Before Getting To Spike	159
The Motor Runs Full Speed In The Wrong Direction	159
SpikeMark Not Responding/ Running Slowly	159
Appendix A - Basic Motion Control Concepts.....	160
Power	160
Position Feedback	160

Motor Tuning..... 161

Over-Travel Limits 161

Index 162

Welcome to SpikeMark

Welcome to SpikeMark, the software designed for theatrical automation. SpikeMark is built to make moving scenery easy and intuitive. As today's shows grow increasingly complex, SpikeMark offers a simple solution to bring the excitement of motorized scenery to every production. Using a Windows® PC and standard Ethernet hardware, you can have a motion control system that is easy to set-up and expand while being incredibly economical. Whether you need to move a deck winch, turntable, roll drop, or any other effect, SpikeMark allows you to focus on the custom aspects of your production by providing a standard way to control motion.

A Word About Safety

The spectacle created by motorized scenery is fantastic. Motion control allows huge scene changes to operate smoothly in a matter of seconds. But this power deserves the utmost caution.

Unlike operating a computerized lighting console or sound system, an automation system can pose a serious danger to performers and technicians. Large pieces of scenery moving at high speed can easily injure, even fatally, any person that is unwittingly caught in its path. Therefore, a large responsibility is placed on you, as the automation operator, to be alert and aware of the people that may be in harm's way.

NOTE: The **GO** button demands a high level of respect.

SpikeMark assists the operator in maintaining safety by providing constant information of every motor's position and status. A hard-wired emergency stop system—Showstopper—provides an immediate way for the operator to shut down all the motion on the stage. Ultimately, it is the operator's judgment and ability to see any potential hazards that will keep the show safe.

System Requirements

To effectively use SpikeMark you will need:

- A Windows® compatible PC (1.2 GHz or faster)
- A dedicated graphics processor (recommended)
- Windows® XP SP2 or better (Windows® 7 recommended)
- Microsoft's .Net 3.5 framework
- 10MB of free storage space
- A network interface card (or built-in networking)
- An Ethernet hub
- Ethernet cables (1 for every device on the network)
- One or more Stagehand motor controllers with electric motors (winches, turntables, etc.)
- A Showstopper emergency stop base station

Installing SpikeMark (in Windows 7)

NOTE: If you already have SpikeMark installed on your computer, you must first **UNINSTALL** the version of SpikeMark that is currently installed [as described in Step 1, A through E, below].

1. Steps A through E are only necessary if you have another version of SpikeMark currently on your computer:
 - A. Click on the **Start Menu**.
 - B. Select **Control Panel**.
 - C. Click **Uninstall a Program**.
 - D. Select the **SpikeMark** program and click **Uninstall**.
 - E. Click **Yes** to confirm that you wish to uninstall.
2. Place the SpikeMark installation CD into your disc drive.
3. In the **AutoPlay** window, click on **Open folder to view files** (Figure 1).

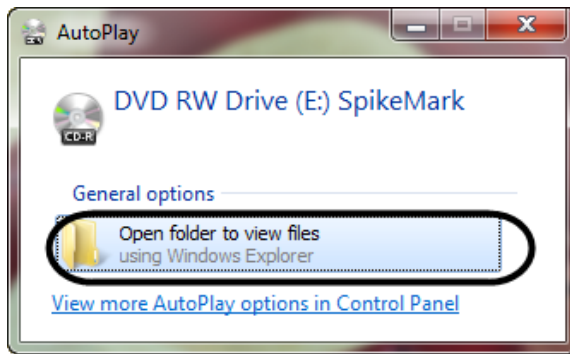


Figure 1

4. Double-click the **SpikeMarkSetup** file (Figure 2).

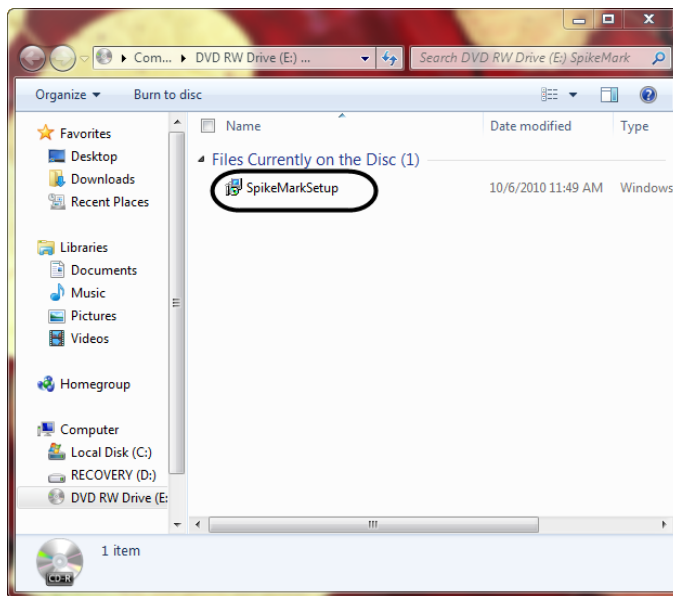


Figure 2

5. In the **Confirm Installation** window, click **Next** (Figure 3).

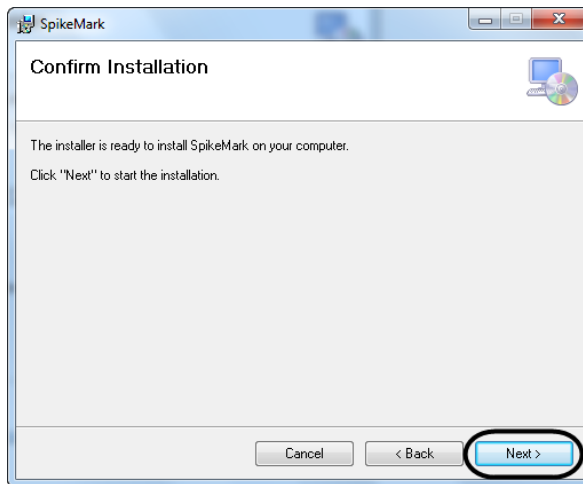


Figure 3

6. In the **Welcome to SpikeMark Setup Wizard** window, click **Next** (Figure 4).

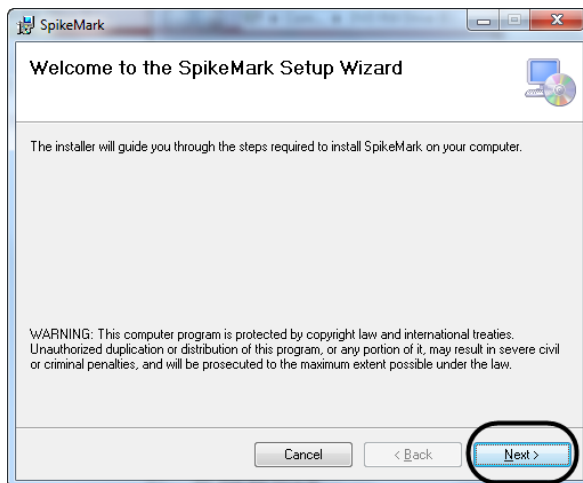


Figure 4

7. In the **License Agreement** window, read the agreement, select **I Agree** and click **Next** (Figure 5).

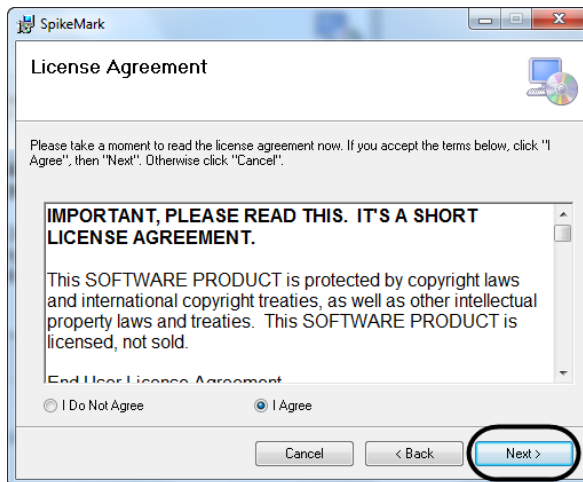


Figure 5

8. The **Installing SpikeMark** window will launch (Figure 6).

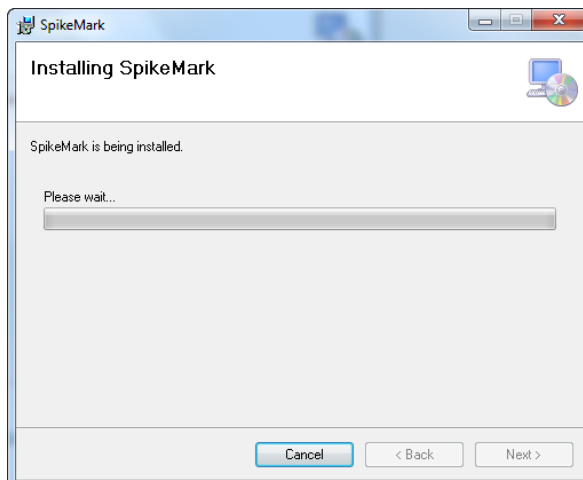


Figure 6

9. In the **Installation Complete** window click **Close** (Figure 7).

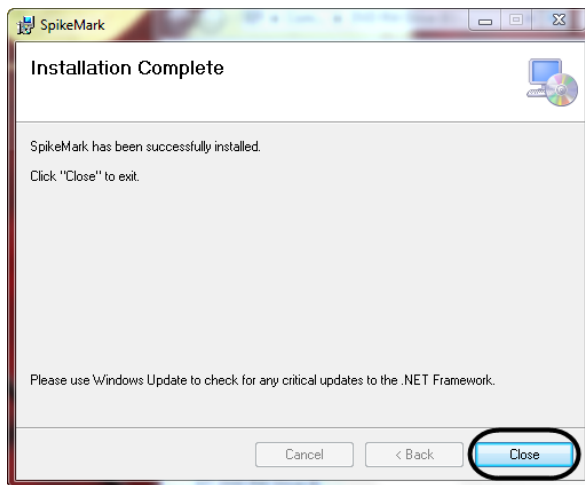


Figure 7

NOTE: If Microsoft's .Net 3.5 framework is not installed on your system, the installer will direct you to a Microsoft update page where the framework can be downloaded. Also, check **Windows Update** regularly to keep your Windows operating system up to date.

10. After the SpikeMark installation is complete, launch the SpikeMark program and enter the **User** and **Password** in the **Registration** window (Figure 8).
*The **User** and **Password** are provided on the sleeve of your SpikeMark installation CD.*



Figure 8

11. Click **OK** in the **Thank you** dialog box (Figure 9) to get started with the SpikeMark program.

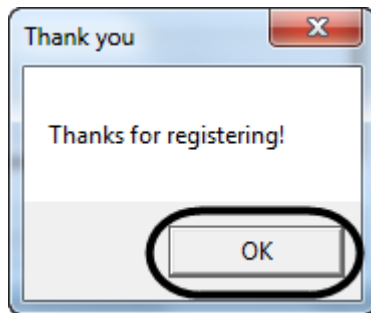


Figure 9

Online Tutorials

There is a selection of video tutorials on the Creative Conners website that cover all the basic uses of SpikeMark. <http://www.creativeconners.com/pages/training.html>

SpikeMark's Newest Features

SpikeMark has a number of great new features, improving on Creative Conner's legacy of providing the easiest software to use for stage automation. SpikeMark's name also highlights one its handiest new features: Spikes. The following are all new SpikeMark features.

Spikes

The **Spikes** feature is designed to mimic the usefulness of real spike tape. Rather than trying to remember bizarre cue position numbers (such as 251.13"), you can now assign names to commonly used positions ('on stage' = 251.13"). When writing cues, you can select a position from the motor's defined spikes (Figure 10).

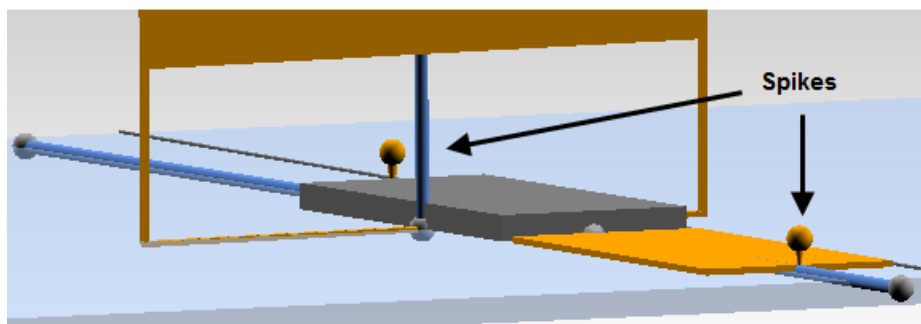


Figure 10

Often during the rehearsal process, the spike positions need to be tweaked. If you were physically pushing the wagons around, you would just move the spike tape on the stage floor to a new position. Similarly, with the new **Spikes** feature, you simply edit the

position of the spike in one place and all cues that use the spike are automatically updated, saving you from manually tracking changes through the show.

Time-Based Cues

Cues can now be written using *time* parameters in addition to speed and acceleration values. You can write cues by specifying how long the cue should run, and how long it should take for the motors to ramp up to speed (Figure 11). Quickly writing cues that match the timing of the show helps reduce tech-table stress.



Figure 11

Streamlined Interface

SpikeMark's interface is now easier to navigate (Figure 12) and has "Splitter Panes" (Figure 13), which allow you to create a computer workspace that works best for you.

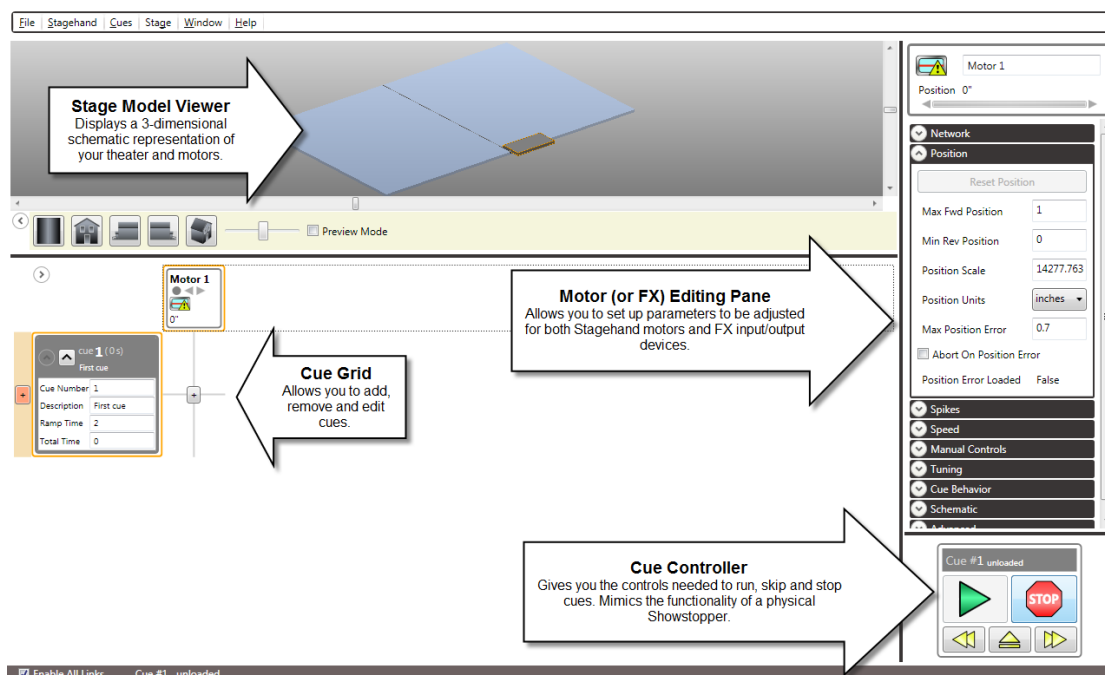


Figure 12

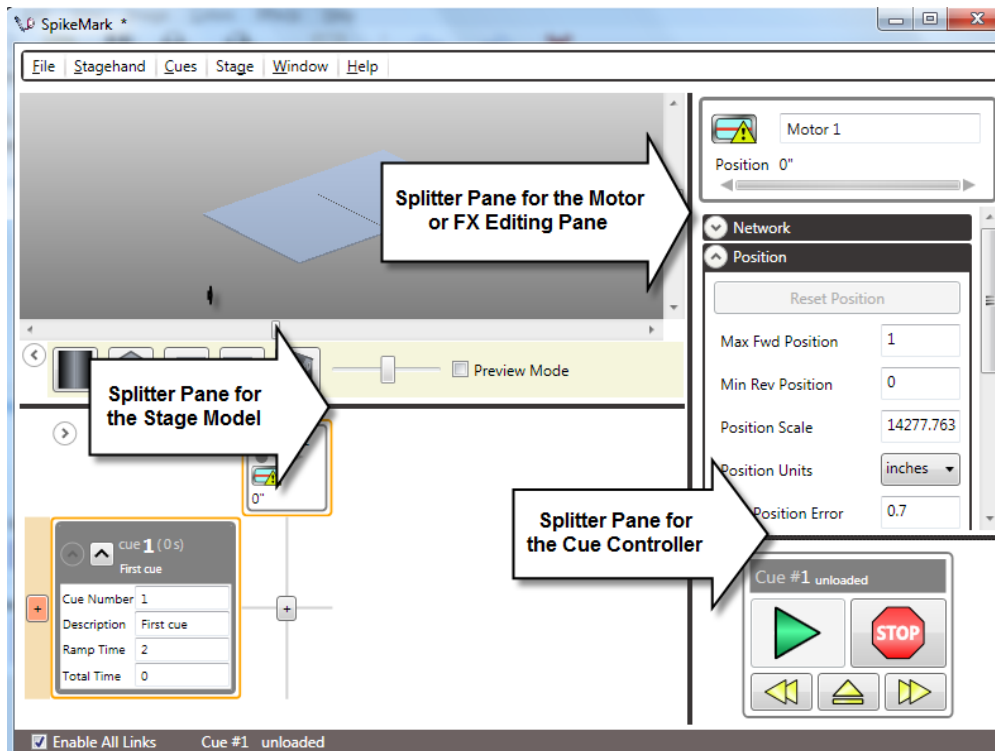


Figure 13

3-Dimensional Stage View

The SpikeMark **Stage Model Viewer** gives a three-dimensional view of the theatre space (Figure 14). Wagons and turntables can be rotated along any axis, making it easy to view lifts and deck winches in a more realistic fashion.

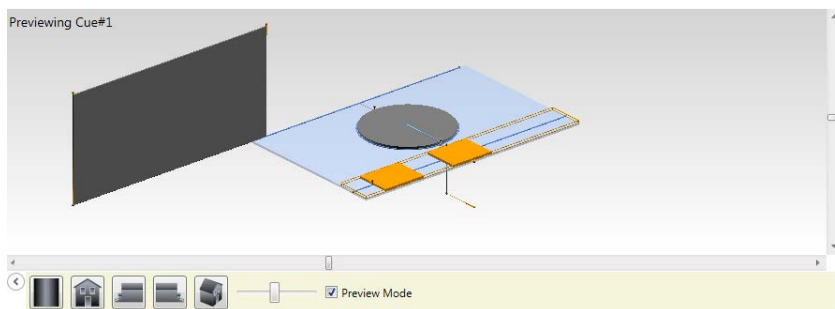
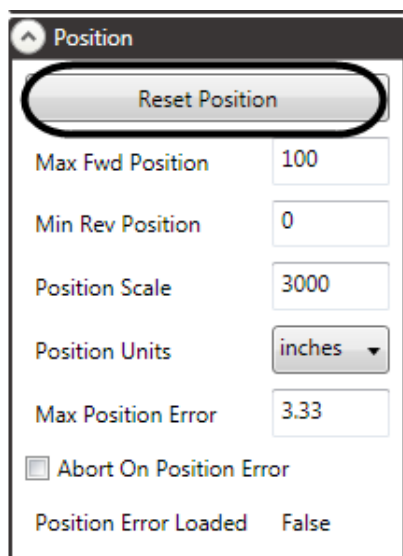


Figure 14

The new **Spikes** feature is integrated with the **Stage Model Viewer** to make it possible to build cues visually and accurately.

Reset Position

The **Reset Position** feature (Figure 15) allows you to reset a motor to an arbitrary position, rather than only being reset to zero. This is helpful if you need to swap out a Stagehand mid-show and need the new Stagehand to pick up the motor position exactly where the old Stagehand left off.



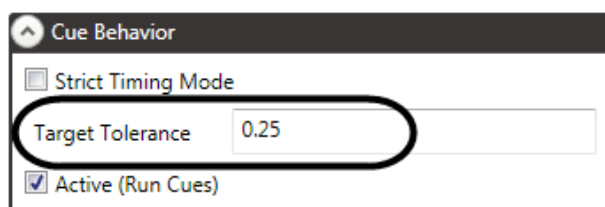
The 'Position' window contains the following controls:

- Reset Position**: A button at the top, circled in red.
- Max Fwd Position**: A text input field with the value '100'.
- Min Rev Position**: A text input field with the value '0'.
- Position Scale**: A text input field with the value '3000'.
- Position Units**: A dropdown menu currently set to 'inches'.
- Max Position Error**: A text input field with the value '3.33'.
- Abort On Position Error**: An unchecked checkbox.
- Position Error Loaded**: A label followed by the value 'False'.

Figure 15

Target Tolerance Mode

Motors can now optionally run their cues in **Target Tolerance Mode** (Figure 16) which makes precise motor tuning less important. When run in this new mode, the motor will continue to try to reach target even if the tuning parameters are too sloppy to get to the target position in the precise time requested. You can enter a tolerance value that specifies when the motor should consider its position 'good enough' (.25", .5", etc.).



The 'Cue Behavior' window contains the following controls:

- Strict Timing Mode**: An unchecked checkbox.
- Target Tolerance**: A text input field with the value '0.25', circled in red.
- Active (Run Cues)**: A checked checkbox.

Figure 16

Stagehand FX Support

In addition to controlling motors, you can now control simple switched devices with the Stagehand FX. When using the Stagehand FX with SpikeMark, each device gives you four switchable outputs and four readable inputs. Outputs can be used to turn simple

devices on & off (lights, valves, small motors, etc.). Inputs can be used as cue triggers for complex cue sequences or to execute operator-defined actions, such as stopping individual motors.

Quick Start

The best way to learn to use SpikeMark is to start using it, so let's jump in and get something moving!

Setting Up Physical Connections

Before anything will work, all of your cables should be connected, as illustrated in Figure 17.

NOTE: Make sure to set the **physical** limits on your motor[s] **before** running the motor with SpikeMark.

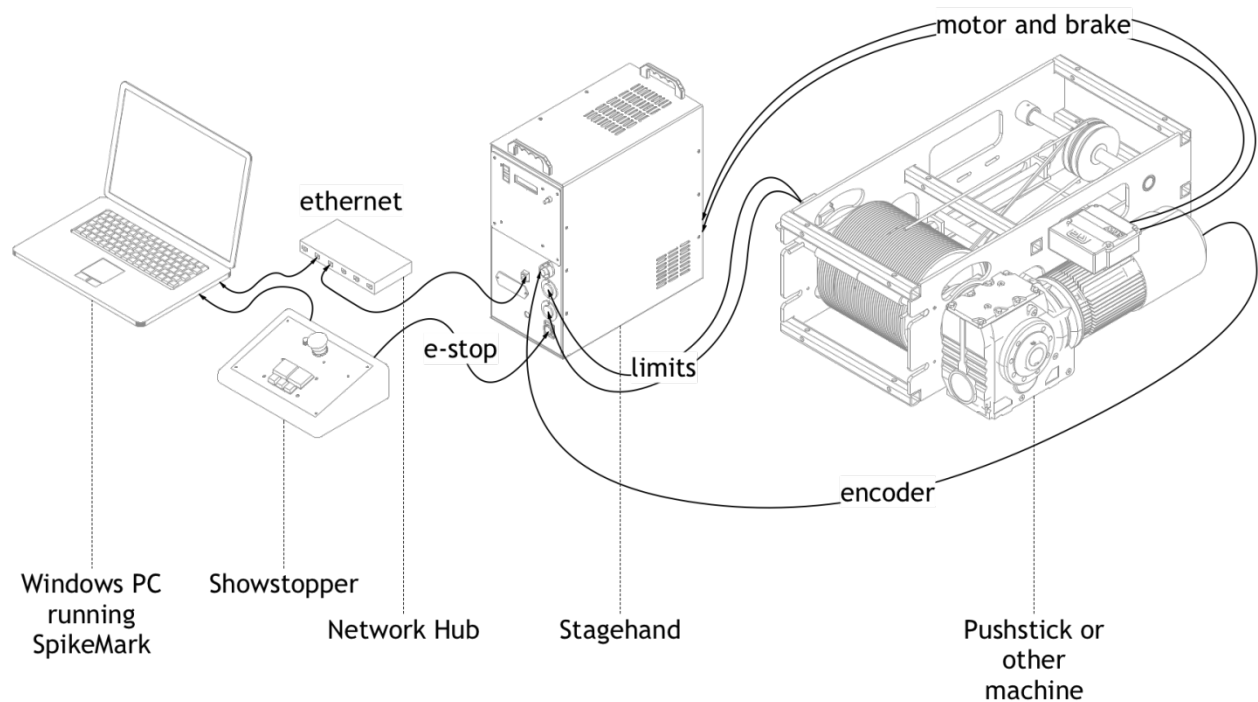


Figure 17

Setting Up a Network Connection (in Windows 7)

Once everything is plugged in, we need to test that the connections work. We'll start by confirming that the Ethernet link works. SpikeMark and Stagehands communicate through static IP addressing. This means that your computer and all Stagehands need to have unique addresses. Much like a postman delivering letters to homes with address numbers nailed to the mailbox, SpikeMark sends its commands to Stagehands with unique address numbers.

IP addresses are made up of four segments separated by a period (for example, "192.168.10.20"). The only trick to setting up a static IP network with SpikeMark and Stagehands is that the first three segments should always match. The last segment number needs to be unique (Figure 18).



Figure 18

Setting the IP Address of Your Computer

1. From the **Start Menu**, select **Control Panel** (Figure 19).

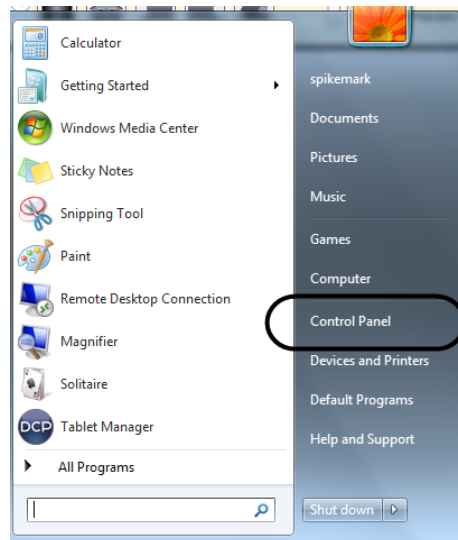


Figure 19

2. Click on the **Network and Internet** link (Figure 20).

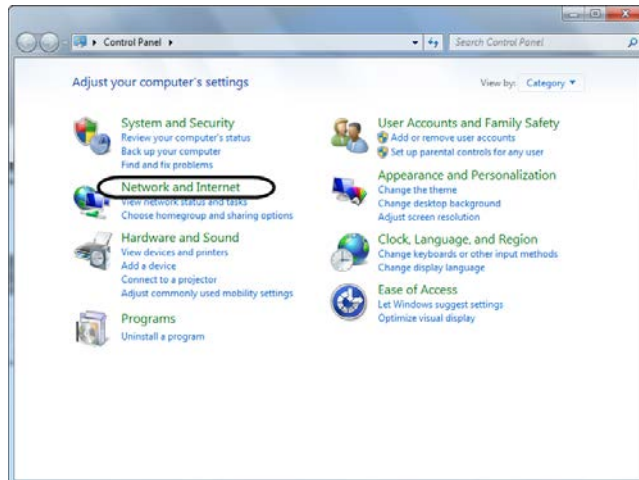


Figure 20

3. Click on the **Network and Sharing Center** link (Figure 21).

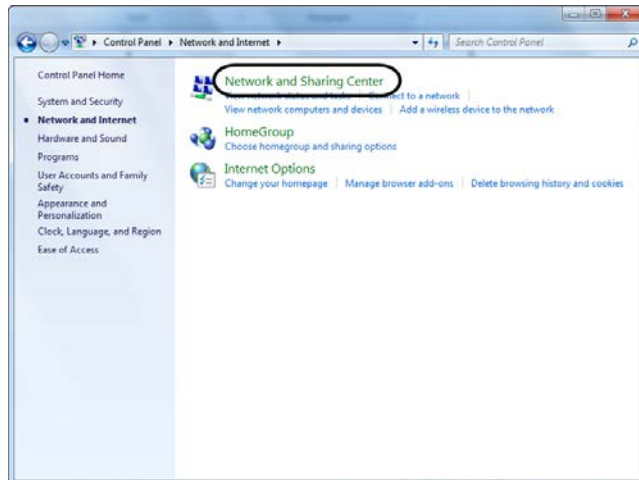


Figure 21

4. Click on the **Local Area Connection** link (Figure 22).

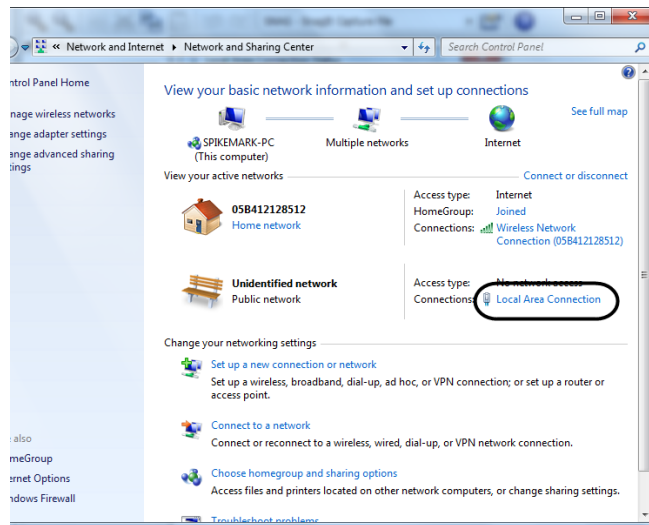


Figure 22

5. In the **Local Area Connection Status** dialog box, click the **Properties** button (Figure 23).

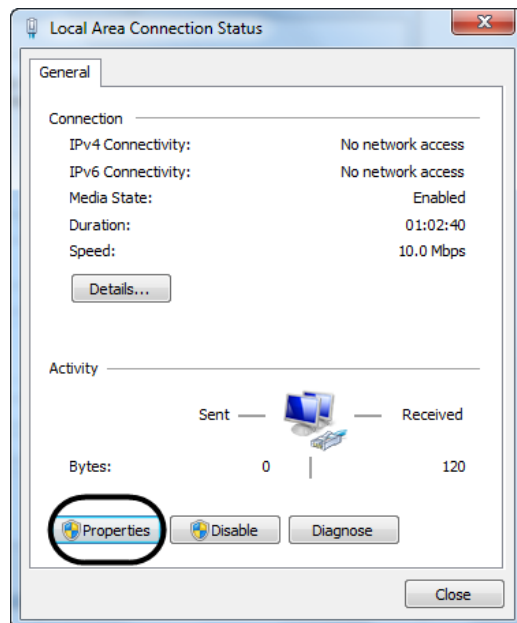


Figure 23

6. Highlight **Internet Protocol Version 4 (TCP/IPv4)** and then click **Properties** (Figure 24).

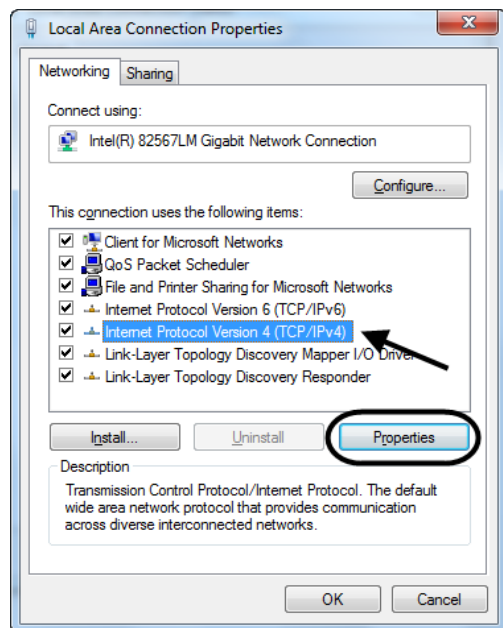


Figure 24

7. In the **Internet Protocol Version 4 (TCP/IPv4) Properties** dialog box, select **Use the following IP address** field and then enter the correct **IP Address** (Figure 25).

NOTE: The first 3 segments of the **IP address** must be shared by your computer and any Stagehand(s). The last segment of the **IP address** **MUST** be unique to each Stagehand and your computer. The **IP address** of your Stagehand(s) is displayed on the face panel.

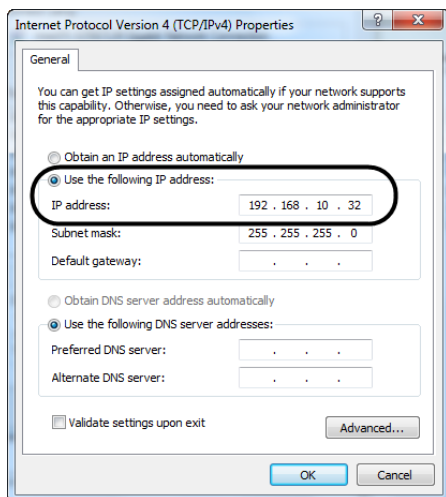


Figure 25

8. Click **OK** to accept the new **IP address**.
9. Click **Close** in the **Local Area Connection Properties** dialog box.
10. Click **Close** again in the **Local Area Connection Status** dialog box.

Establishing a Connection to a Stagehand Motor or FX

With the network address configured on your computer, we're ready to establish a connection to a Stagehand.

1. Launch SpikeMark.
The various sections of the main screen are labeled in Figure 26.

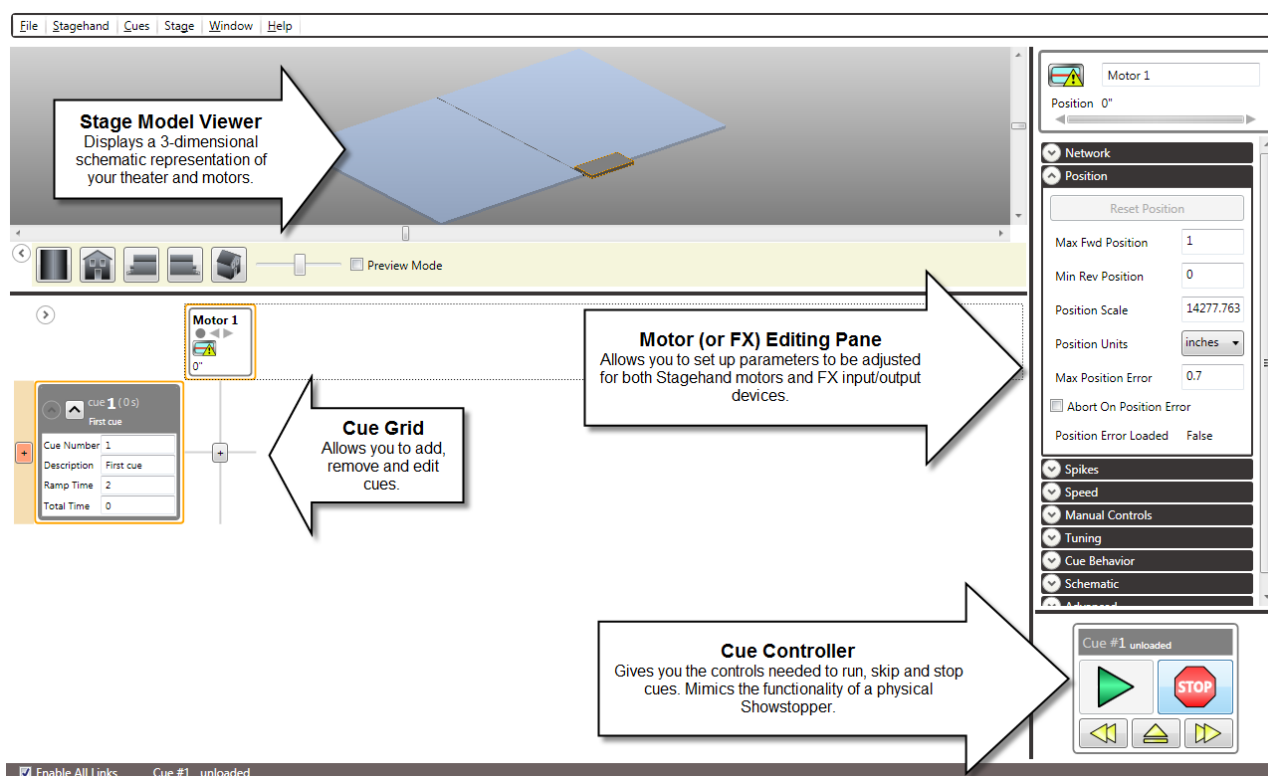


Figure 26

2. Select the motor or FX you wish to connect to by clicking on it in the **Cue Grid**. An orange highlight will appear around the selected motor (Figure 27).

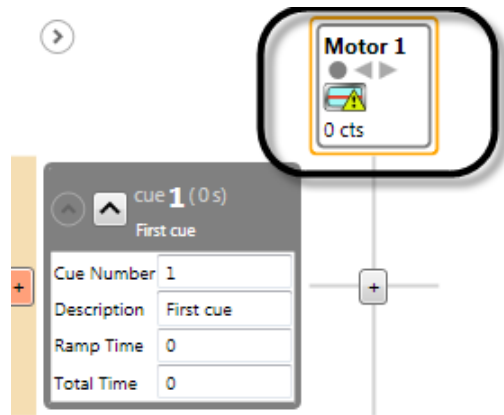


Figure 27

3. In the **Network** section of the **Motor Editing Pane** on the right side of the screen, enter the **IP Address** of your Stagehand Motor or FX and click **Connect** (Figure 28). *In this step, make sure you copy the IP Address exactly as it appears on the face panel of the **Stagehand** LCD display.*

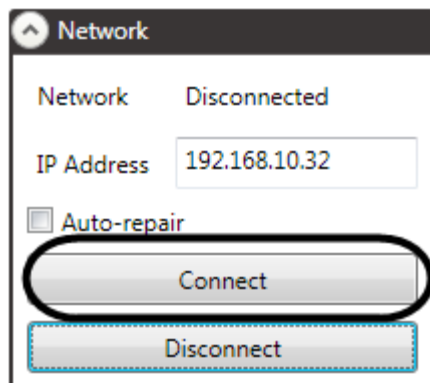


Figure 28

4. The Network status should then change from **Disconnected** to **Connected** (Figure 29). *If there's a network error, check your IP settings on the computer and Stagehand and then check your cabling.*

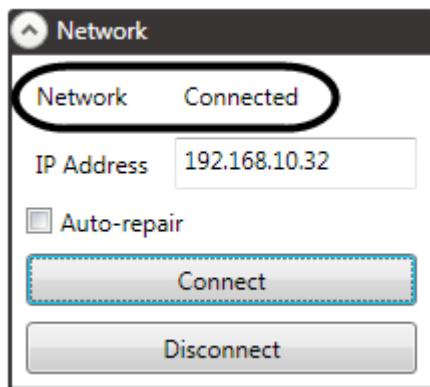


Figure 29

Confirming Encoder Feedback

1. Mark the motor's physical position on the stage with a piece of spike tape.
This will be important in the next section, "Setting the Position Scaling Factor" on page 26.
2. With the motor still selected in the **Cue Grid**, scroll through the **Motor Editing Pane** on the right side of the screen until you see the **Manual Controls** section (Figure 30).

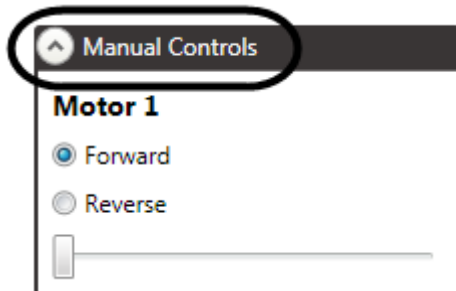


Figure 30

3. Select the **Forward** direction (Figure 31)



Figure 31

4. Make sure that there are no obstructions in the motor's path, and slowly start increasing the speed slider.

You have to hold the mouse button down while the motor is moving. If you release the mouse button, the motor will stop. It's a dead-man switch.

5. Confirm that the position number at the top of the **Motor Editing Pane** is increasing as the motor moves forward (Figure 32).

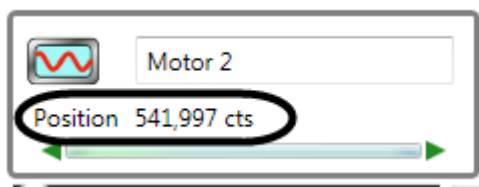


Figure 32

NOTE: The **forward** direction of the motor has to have **increasing** encoder counts. If the encoder counts are decreasing while the motor is moving forward, you need to fix the polarity of the motor. To fix the polarity of the motor, open up the motor plug and swap any two of the power wires or change the encoder wiring to reverse the encoder's polarity. (See **NOTE**, page 45).

DO NOT MOVE THE GROUND WIRE.

Setting the Position Scaling Factor

1. In the **Manual Controls** section of the **Motor Editing Pane**, move the motor as far forward as it can safely travel.
2. Physically measure the distance from the spike tape that you placed on the floor in Step 1 of "Confirming Encoder Feedback" (page 25) to the current position of the motor. *As an example, we'll make the physical measurement equal to 500 inches.*
3. Divide the **Position** number by the number of measured physical inches.
Position = 541,997 encoder counts (Figure 33)
Measured inches = 500 inches (from the example in step 2, above)

$$541,997 \text{ encoder counts} \div 500 \text{ inches} = 1,083.994 \text{ counts per inch.}$$

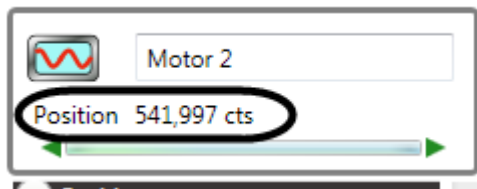


Figure 33

4. In the **Position** section of the **Motor Editing Pane**, enter the computed number into the **Position Scale** field (Figure 34).

Position Scale = 1,083.994 counts per inch (the answer from step 3, above)

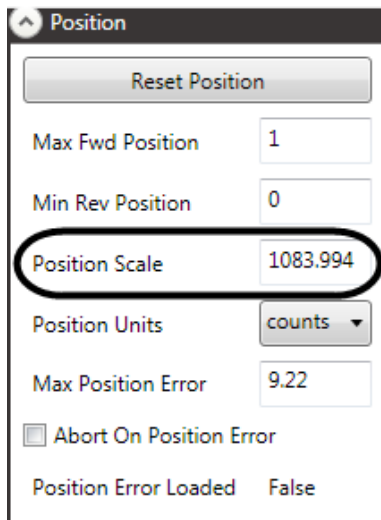


Figure 34

5. Select **"inches"** from the **Position Units** drop-down box (Figure 35).

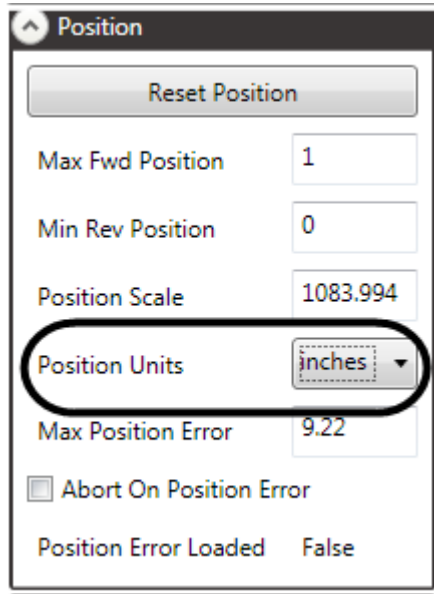


Figure 35

6. The **Position** now displays a position of **500"** (Figure 36).

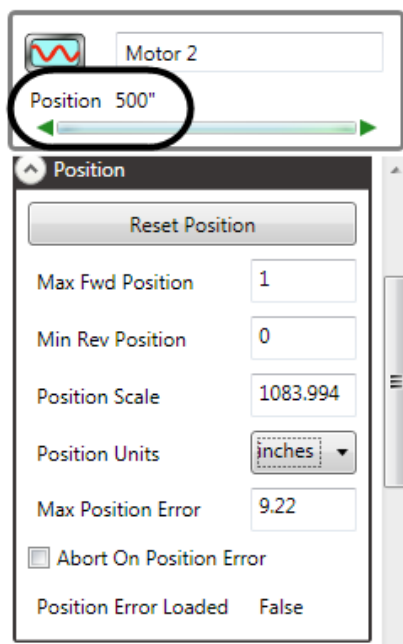


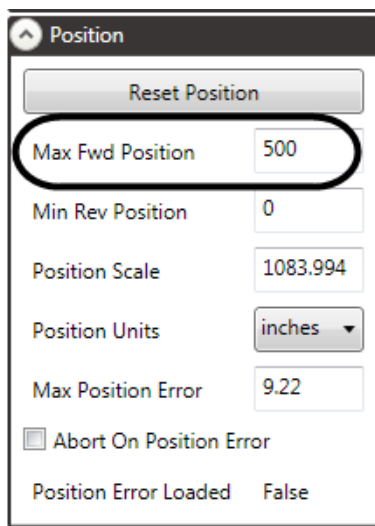
Figure 36

Setting Soft Limits

1. In the **Position** section of the **Motor Editing Pane**, enter "0" for the **Min Rev Position**. *This will prevent you from accidentally writing cues less than 0.*

2. In the **Position** section of the **Motor Editing Pane**, enter your current motor position as the **Max Fwd Position**, which will prevent you from writing cues greater than the current motor position.

*In Figure 37, the **Max Fwd Position** is “500,” which was established in Step 2 of the “Setting the Position Scaling Factor” section on page 26.*

The image shows a software interface titled "Position" with a list of motor parameters. A red circle highlights the "Max Fwd Position" field, which contains the value "500". Other fields include "Min Rev Position" (0), "Position Scale" (1083.994), "Position Units" (inches), "Max Position Error" (9.22), and an "Abort On Position Error" checkbox. At the bottom, it says "Position Error Loaded False".

Position	
Reset Position	
Max Fwd Position	500
Min Rev Position	0
Position Scale	1083.994
Position Units	inches
Max Position Error	9.22
☐ Abort On Position Error	
Position Error Loaded	False

Figure 37

Setting a Motor's Default Speed Parameters

1. In the **Cue Grid**, select the motor.
2. In the **Manual Controls** section of the **Motor Editing Pane**, use the manual controls to start moving in reverse as fast as you are comfortable. While the motor is moving, watch the **Speed** value in the **Speed** section of the **Motor Editing Pane** and mentally note how fast the motor is moving (Figure 38).

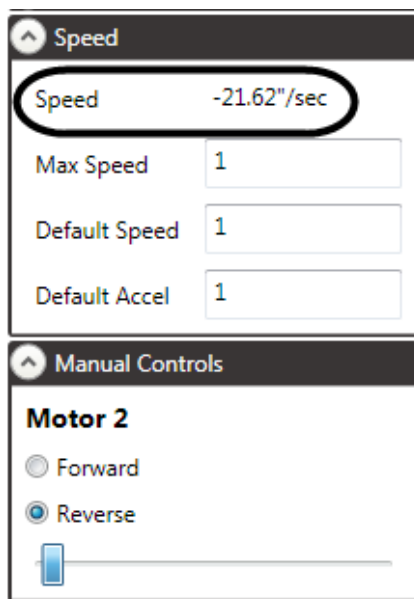


Figure 38

3. Slowly ramp the motor back down to zero speed and then release the mouse button to stop the motor.
4. Enter the speed value you observed in Step 2, above, into the **Max Speed** field in the **Speed** section in the **Motor Editing Pane** (Figure 39).
*In the example below, the **Max Speed** is “21.62”*

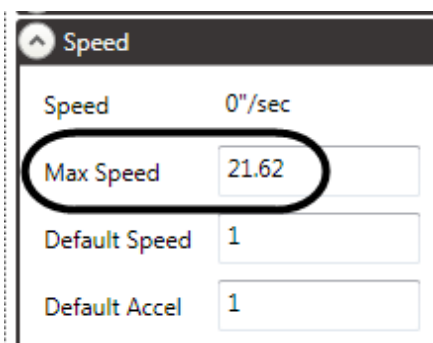
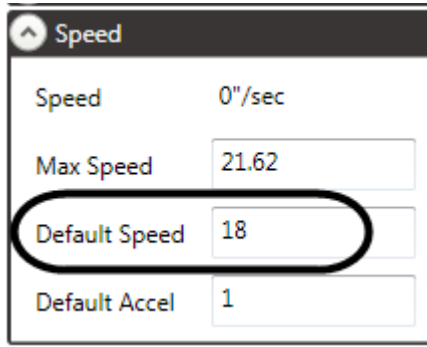


Figure 39

5. Enter a speed value into the **Default Speed** field in the **Speed** section in the **Motor Editing Pane** (Figure 40).
*The **Default Speed** is used when writing new cues. If you have a speed that is commonly used for the motor, using a **Default Speed** is a handy shortcut when writing cues. It's a convenience feature that can be overridden at anytime.*

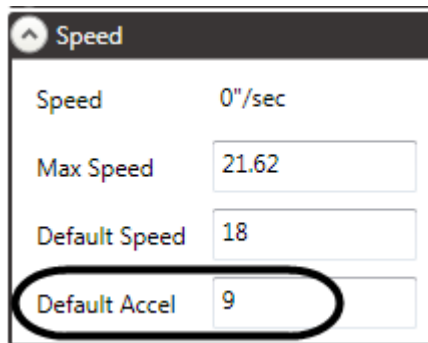


A screenshot of a 'Speed' settings panel. It contains four input fields: 'Speed' (0"/sec), 'Max Speed' (21.62), 'Default Speed' (18), and 'Default Accel' (1). The 'Default Speed' field is circled in black.

Speed	0"/sec
Max Speed	21.62
Default Speed	18
Default Accel	1

Figure 40

6. Enter an acceleration value into the **Default Accel** field in the **Speed** section in the **Motor Editing Pane** (Figure 41).
Acceleration values are given in inches/sec/sec (inches per second per second).



A screenshot of a 'Speed' settings panel, similar to Figure 40 but with the 'Default Accel' field circled in black. The 'Default Speed' field now contains the value 18.

Speed	0"/sec
Max Speed	21.62
Default Speed	18
Default Accel	9

Figure 41

NOTE: As a rule of thumb, it is best to enter a value in the **Default Accel** field that is **half** of the **Default Speed**, which—in the example above—gives you a default acceleration that will last for 2 seconds.

Writing Two Sample Cues

In the following steps, the first cue will send a motor to its Maximum Forward position (**Max Fwd Position**) and the second cue will send it back to its Minimum Reverse position (**Min Rev Position**). There's already a Cue 1 defined by default when you open SpikeMark, so we'll add a Cue ".5," which will allow this cue to start *before* Cue 1.

1. Right-click in the **Cue Grid** and select **Add Cue**.
*You could also press **Ctrl + U**, or from the **Cues** menu, click **Add Cue**.*
2. Enter a **Cue Number** and a **Cue Description** in the **New Cue** dialog box and click **OK** (Figure 42).
Cue numbers can be any positive decimal value (1, 2.34, 1008.38972, etc.); Cue descriptions are labels you can use to help identify what the cue does and use to quickly search for a cue when running a show.

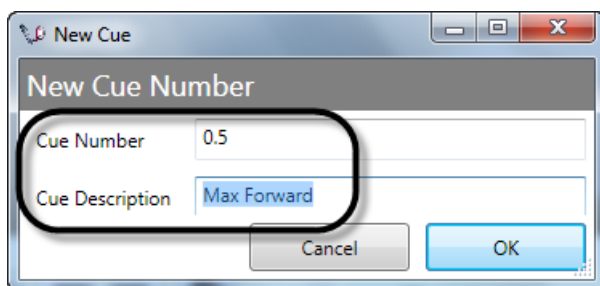


Figure 42

3. Add a movement for **Motor 1** in **Cue .5** by pressing the "+" button in the **Cue Grid** (Figure 43).

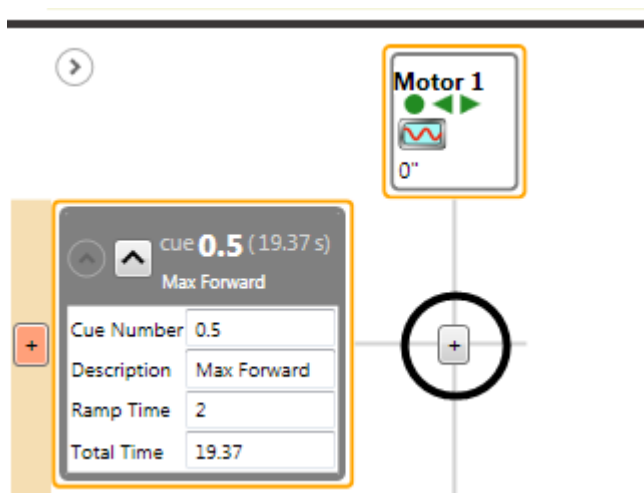


Figure 43

4. Change the **Position** number in the **Cue Grid** to match the value you entered as the **Max Forward Position**. (Figure 44).

*In the example below, the value would be “500.” This value is also held in the **Max Fwd Position** field in the **Position** section of the **Motor Editing Pane**.*

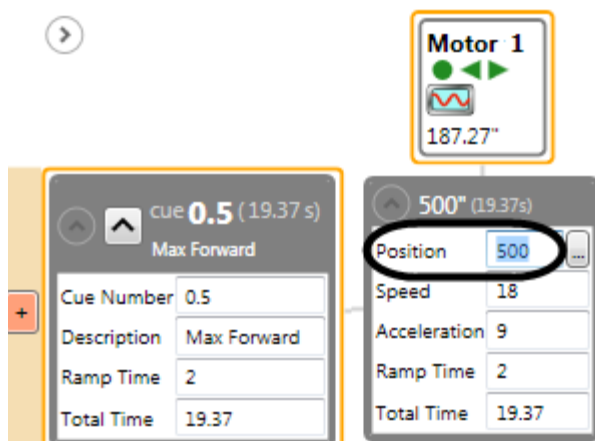


Figure 44

5. Leave the **Speed** and **Acceleration** values as their default values.
6. Now that we've added Cue .5, let's write another Cue. By default, **Cue 1** already exists. Enter its **Description** as "Min Reverse," then repeat steps 3 through 5 and enter a **Position** value of "0" (Figure 45).

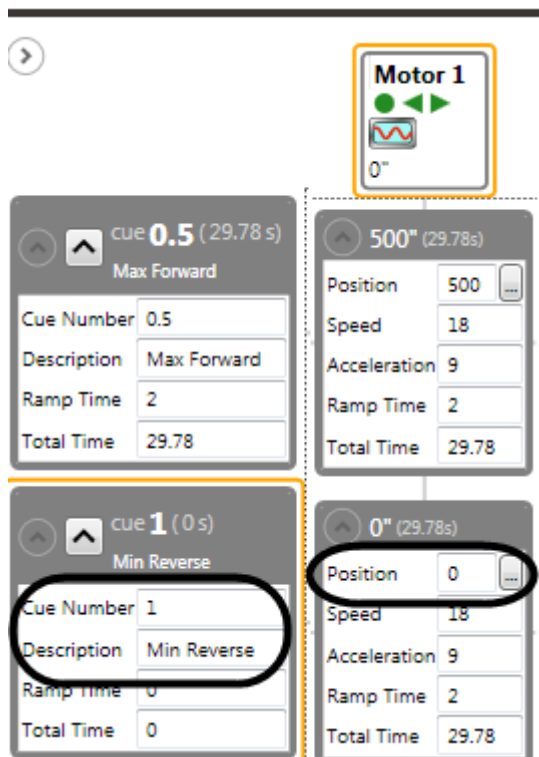


Figure 45

Running the Two Sample Cues

NOTE: Depending on your motor system, the cues may or may not work well at this point. We're going to run the cues with the default motor tuning parameters and hope for the best. If the default tuning isn't good enough, the motor won't complete the cue, it will just sit there struggling at very low speed to complete the cue. Don't worry if your cue doesn't run well yet, we'll fix it with motor tuning in the next section.

1. To load **Cue 1**, press the **Load Cue** button on the cue in the **Cue Grid** (Figure 46).

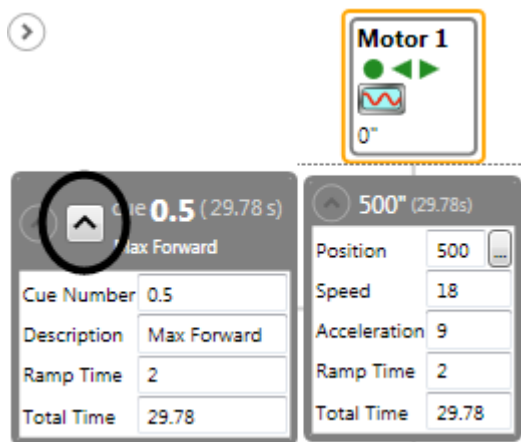


Figure 46

2. The border around the cue box will turn red, indicating that the cue is loaded (Figure 47).

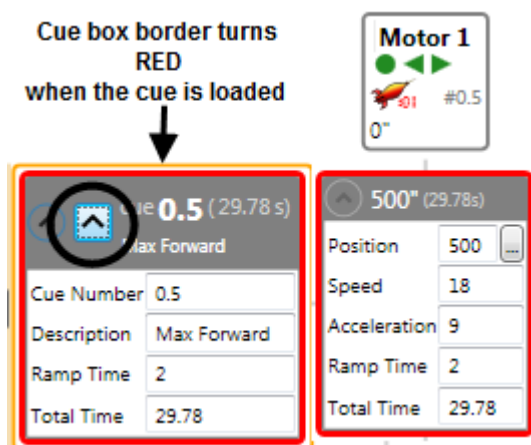


Figure 47

- To run **Cue 1**, press the **Run Cue** button in the **Cue Controller** section on the right lower side of the screen (Figure 48).



Figure 48

- As the cue is running, the border around the cue box in the **Cue Grid** turns green (Figure 49).

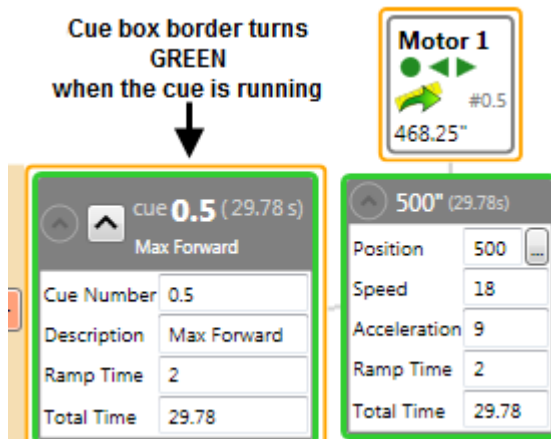


Figure 49

- When the cue completes, the border around the cue in the **Cue Grid** turns blue (Figure 50).

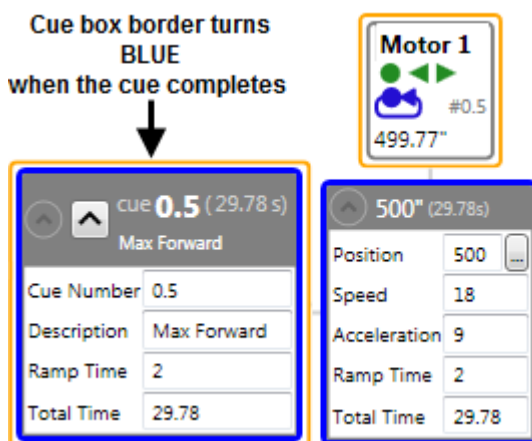


Figure 50

NOTE: If the border stays green, but the motor is no longer moving, this means that the motor tuning needs to be improved. Abort the cue by pressing the **Emergency Stop** button on your Showstopper, skip step 6 below and go to the next section: “Basic Tuning,” below.

6. If the border around the cue box in the **Cue Grid** is blue and has properly completed, then run **Cue .5** by repeating steps 1 through 5 above.

Basic Tuning

If your first attempt at running cues didn’t work and the motor never fully completed the cues, then the tuning parameters need to be adjusted. Tuning can be a complex issue, but we’re going to take a practical approach in this section without getting bogged down in the details. For more information, see the “Motor Tuning” section on page 51.

1. In the **Motor Editing Pane** on the right side of the screen, scroll until you find the **Tuning** section.
2. Increase the **Proportional Gain** value by 1 (Figure 51).

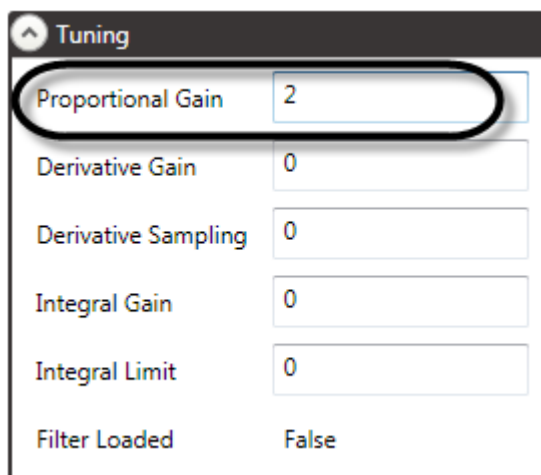


Figure 51

3. Run a Cue to move the motor to the opposite side of the stage by following the instructions in the previous section: “Running the Two Sample Cues.”
4. If the Cue now completes successfully, you are done. Otherwise repeat steps 1 through 3 until the motor repeatedly completes the cues successfully.

SpikeMark's Main Screen

The main screen of the SpikeMark interface is composed of four areas (Figure 52, clockwise, from top left):

Stage Model Viewer: The upper left area, which contains and displays a 3-dimensional schematic representation of your theatre and motors.

Motor Editing Pane: The far right area, which allows setup parameters to be adjusted for a Stagehand Motor.

Cue Controller: The bottom right area, which mimics the functionality of a physical Showstopper, giving you the controls needed to run, skip, and stop cues.

Cue Grid: The lower center area, where cues can be added, removed, and edited.

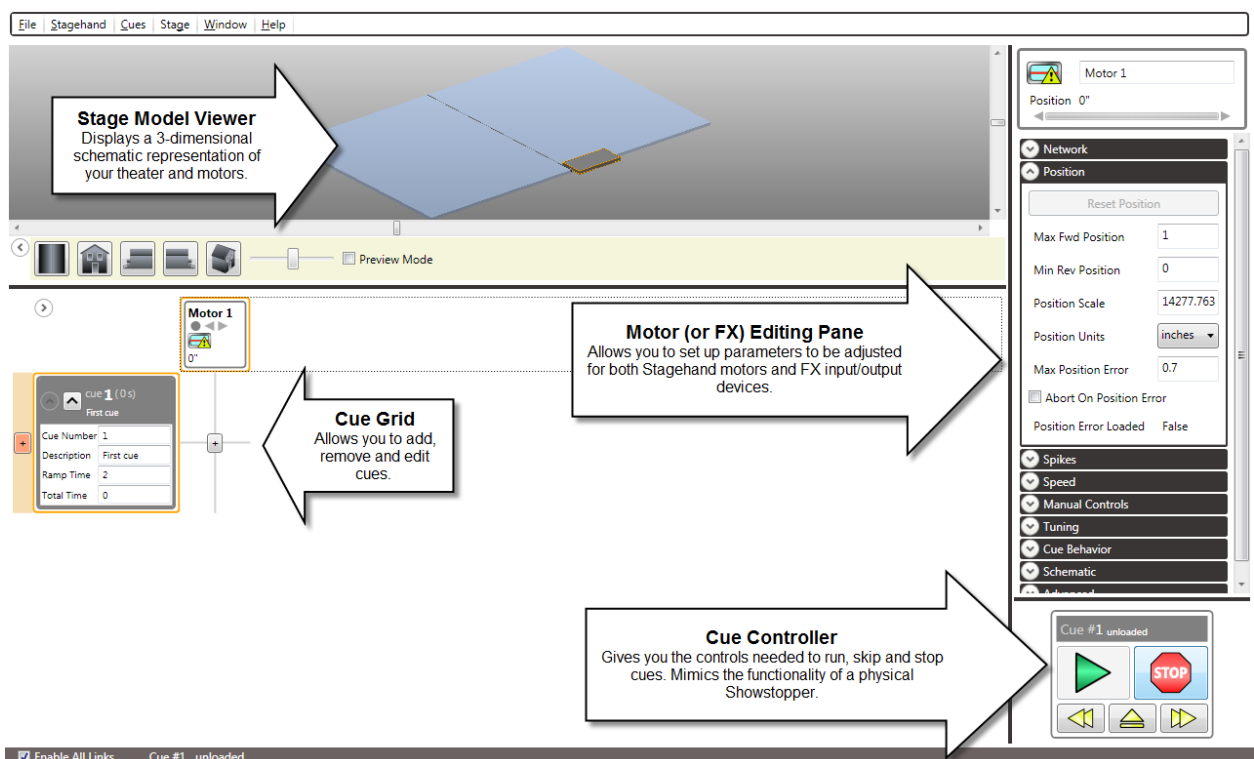


Figure 52

FX Editing Pane: The far right area, which allows setup parameters to be adjusted for a Stagehand FX input/output device (Figure 53).

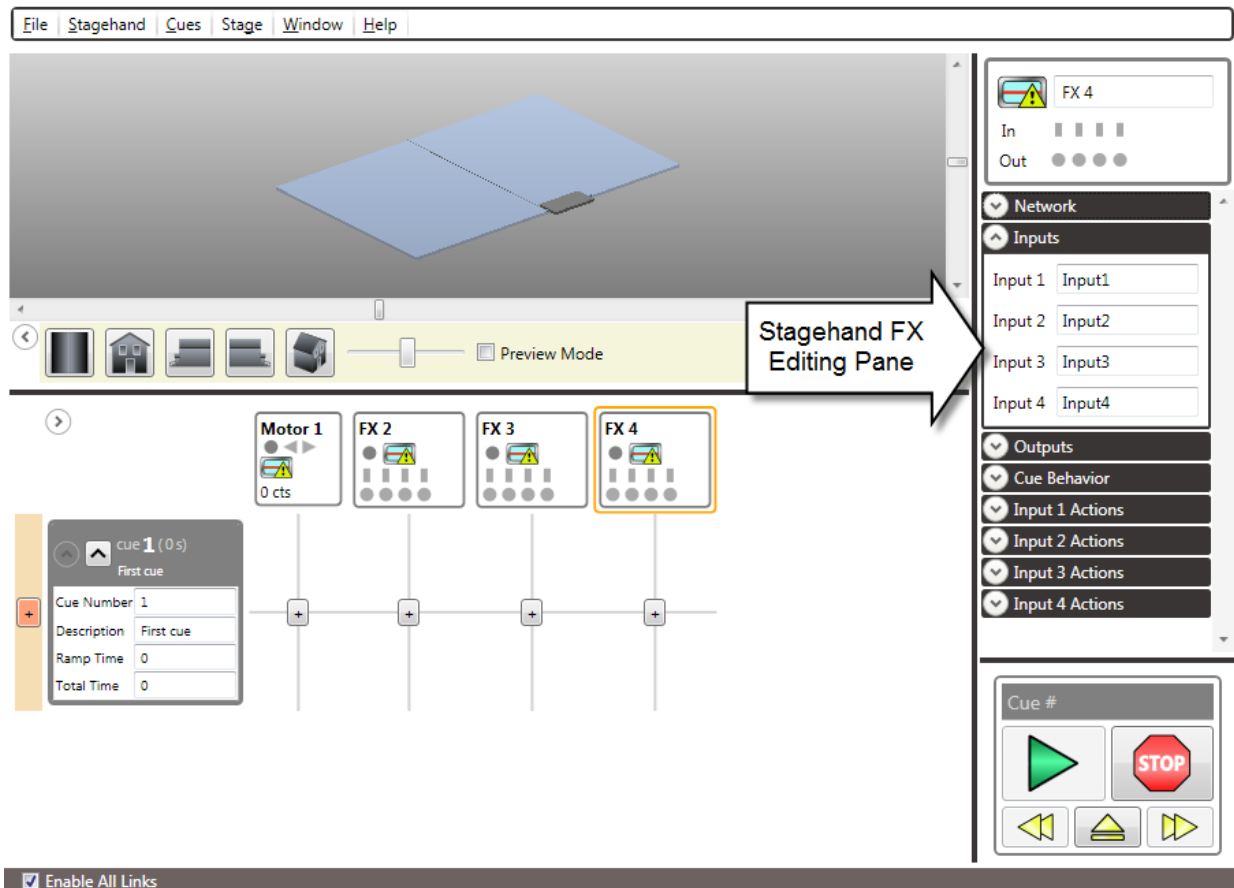


Figure 53

Splitter Panes allow you to adjust the size of the four sections of the SpikeMark screen (the **Stage Model Viewer**, the **Motor or FX Editing Pane**, the **Cue Controller** and the **Cue Grid**). Click and drag on the **Splitter Panes** that separate each section (Figure 54). To fully collapse a pane, double-click on the **Splitter Pane**. To expand a collapsed pane, double-click again on the **Splitter Pane**.

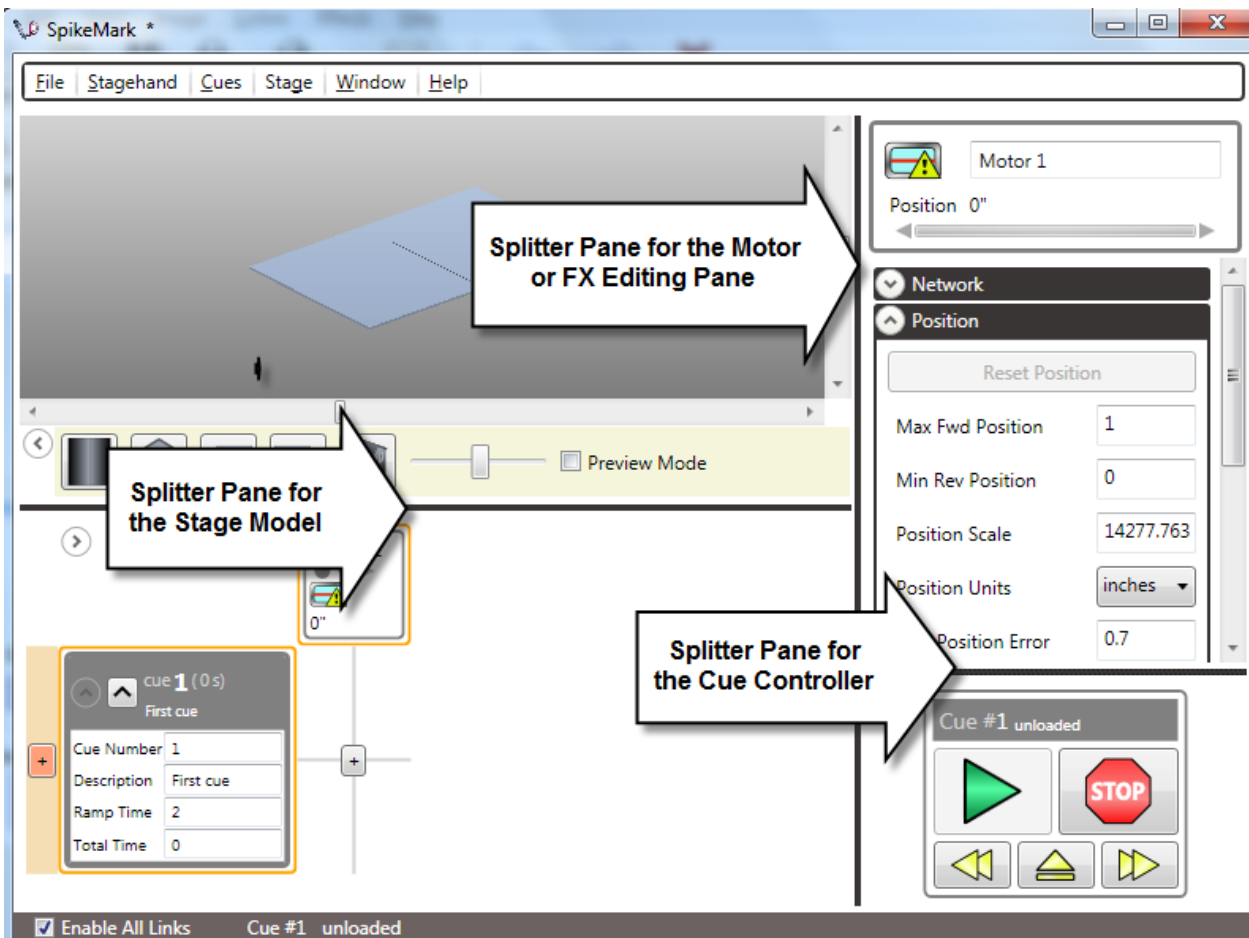


Figure 54

The Stagehand Motor

Adding a New Stagehand Motor

1. Right-click in the **Cue Grid** and select **Add Motor** (Figure 55).

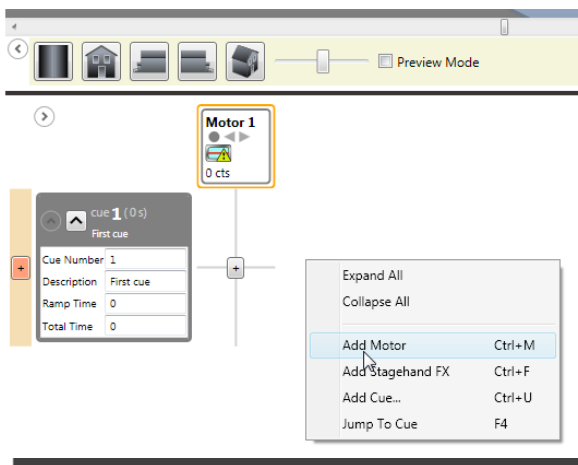


Figure 55

Or in the **Stagehand** menu, click **Add Motor** (Figure 56).

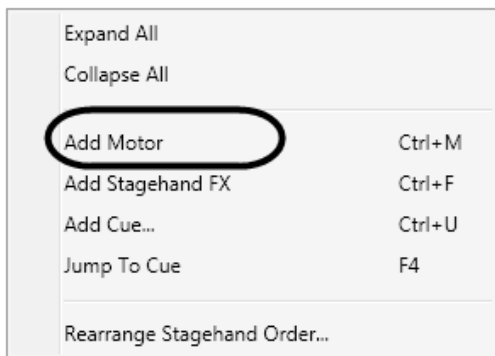


Figure 56

Or, press **Ctrl + M**.

Stagehand Motor Configuration

Each Stagehand Motor in a show has a collection of parameters that are typically set *once* during a production. Parameters include properties such as the name of the motor, the IP

address of the motor, the motor's maximum speed, etc. The **Motor Editing Pane** on the right side of the screen allows you to edit all of the configuration parameters for any motor in the show.

To edit a Stagehand Motor's configuration parameters, select the motor in the **Cue Grid** by clicking on it. The selected Stagehand Motor is then highlighted in orange (Figure 57).

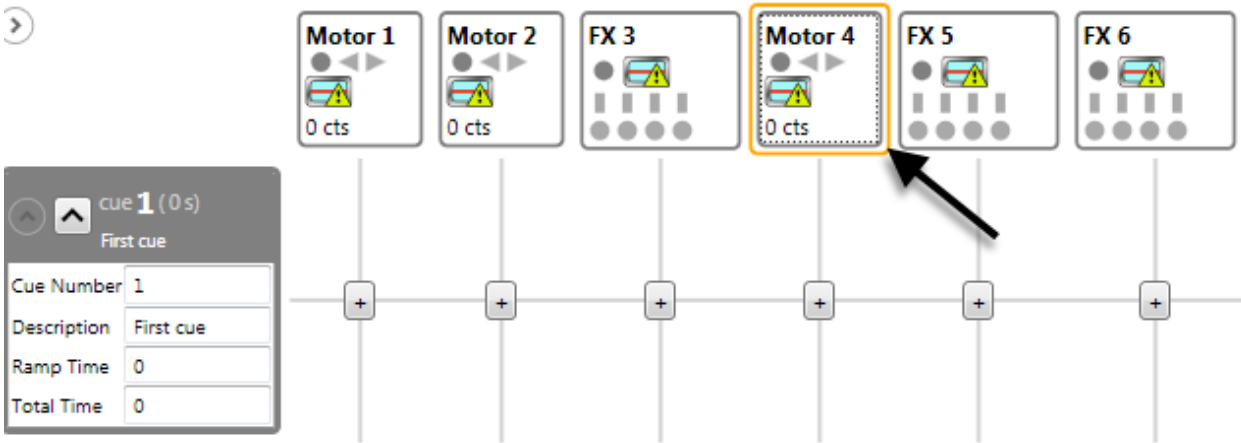


Figure 57

Once you have selected a motor in the **Cue Grid**, you can begin editing its parameters in the **Motor Editing Pane**. The **Motor Editing Pane** is a scrolling window that contains collapsible windows with each section grouped by function (such as Network, Position, Spikes, etc.). You can collapse or expand each section by clicking the up/down carrot top arrow in order to show or hide each section's detailed information (Figure 58).

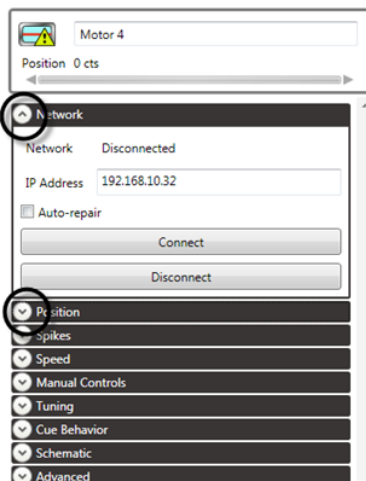


Figure 58

Renaming a Motor

To help you identify all the motors in your show, you can give each motor a name. Typically, motor names reflect the *purpose* of the motor and makes writing cues more intuitive.

1. Select the motor to be edited in the **Cue Grid** and then click into the field in the **Motor Editing Pane** that contains the default motor name.

In the example below, the default name is “Motor 4” (Figure 59).



Figure 59

2. Type a new name.

In the example below, the new name is “Stage Right Winch” (Figure 60).



Figure 60

3. Press the **Tab** key or click out of the field, to update the name. The new motor name will be updated in the **Cue Grid** (Figure 61).

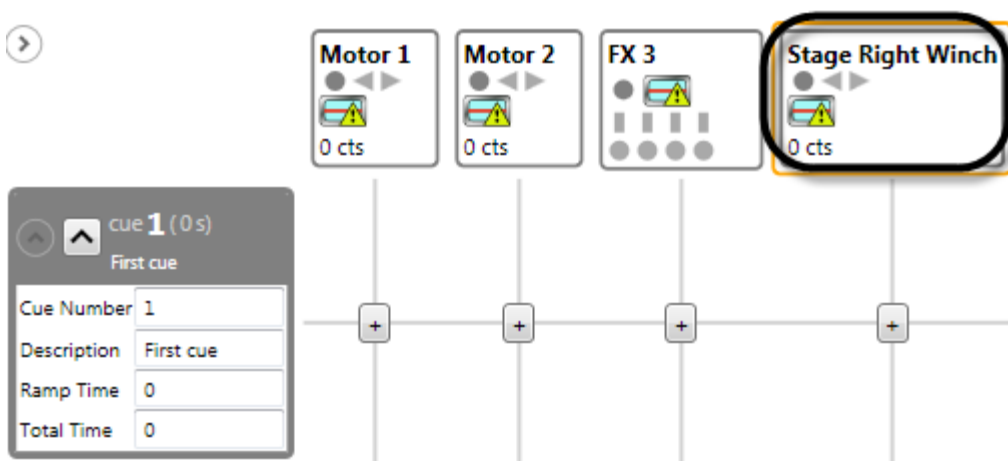


Figure 61

Motor Status

Motor status information is displayed next to the motor name in the **Motor Editing Pane** and in the **Cue Grid** column headers. Each icon status is described below.

Icon	Status
	Motor disconnected from the network.
	Motor connected to the network, but idle.
	Motor is inactive and won't run cues or affect cue links
	Motor has completed a cue.
	Emergency stop.
	Cue loaded.
	Soft stopping, but not yet fully stopped.
	Stopped mid-cue by a soft-stop command.
	In manual mode with the motor enabled, but zero-speed.
	Moving forward in manual mode.
	Moving reverse in manual mode.
	Running a cue forward in target-tolerance mode.
	Running a cue forward in strict-timing mode.
	Running a cue reverse in target-tolerance mode.
	Running a cue reverse in strict-timing mode.
	Stopped by a forward limit switch on the motor.
	Stopped by a reverse limit switch on the motor
	Stopped by a position error fault.

Manually Moving a Motor Using the On-Screen Controls

NOTE: You will frequently need to move a motor forward or backward without running a cue. During initial setup, tech, and work calls the scenery often needs to be nudged around the stage. You can either run the motor using the **physical jog buttons** on the **Stagehand**, or you can use SpikeMark's on-screen **Manual Controls** to move the motor, as described in the steps below.

1. Select the motor you wish to move in the **Cue Grid**.
2. Confirm that the motor is connected to the network. If it's not connected, establish a network connection (see "Setting Up a Network Connection" on page 19).
3. In the **Motor Editing Pane** on the right side of the screen, scroll to the **Manual Controls** section (Figure 62).

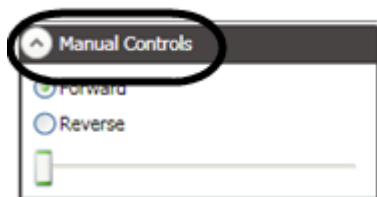


Figure 62

4. Select a direction, either **Forward** or **Reverse** (Figure 63).

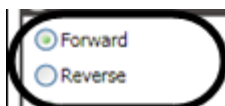


Figure 63

5. Click and drag the slider slowly to the right, which will release the brake and ramp up the motor speed (Figure 64).

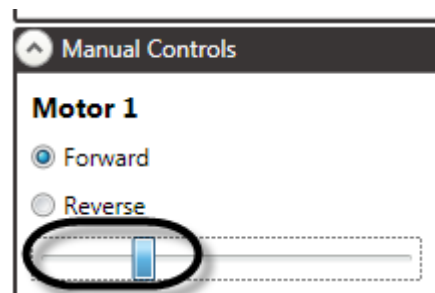


Figure 64

6. To stop the motor, slowly move the slider to the left and then release the mouse button. The motor will stop and the brake will engage.

NOTE #1: While **manually jogging** a motor in the **Forward** direction, the encoder counts should be **increasing**.

While manually jogging the motor in the **Reverse** direction, the encoder counts should be **decreasing**.

If this is NOT the case, you should correct the wiring by **swapping** the A encoder wire w/ the B encoder wire and swapping the /A encoder wire w/ the /B

NOTE #2: While **manually jogging** a motor **forward**, if the machinery is traveling in the **wrong** direction [for example, you want a piece moving forward ONSTAGE, but it's moving OFFSTAGE] and the encoder counts are decreasing, then you need to swap the two power wires so that it goes in the correct direction and the counts increase.

If NOTE #2 applies to a **DC motor**: switch the A^+ wire with the A^- wire.

If NOTE #2 applies to an **AC motor**: switch any **two** of the three **power** wires

Do NOT switch the ground wire!

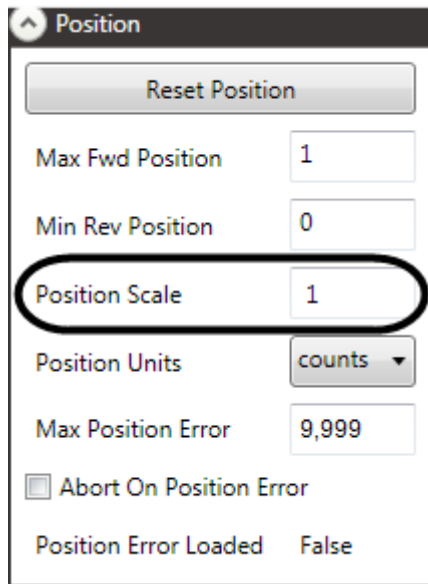
Scaling Motor Position

Whenever a piece of scenery moves, an encoder connected to that scenery spins. As the encoder spins, it generates a stream of electrical pulses. The Stagehand connected to the encoder counts those pulses to determine where the scenery actually is. The Stagehand relays the position back to SpikeMark, which then displays the position information on screen. By default, SpikeMark displays the information as raw encoder counts. Viewing the position of the scenery as raw encoder counts is almost never ideal. Usually you'll want to see the motor's position displayed in something more meaningful, such as inches or degrees.

Motor scaling allows you to specify a ratio between raw encoder counts and physical units. With a position scale defined you can view the motor's current position and write cues in a more user-friendly manner. SpikeMark can display position as inches, feet, millimeters, meters

or degrees. Therefore, rather than having to type 1,245,120,295 encoder counts, you would simply enter 123.5 inches.

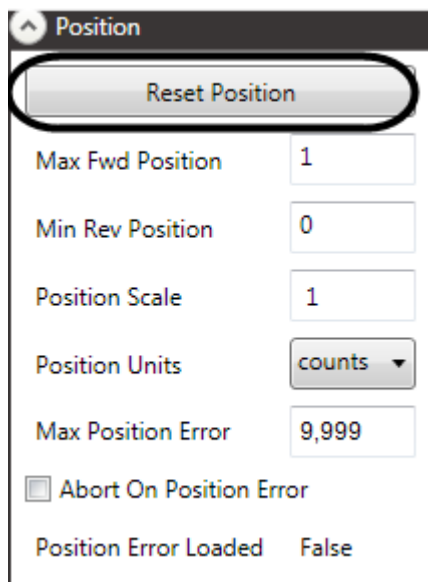
1. Confirm that the current **Position Scale** field is set to “1” in the **Position** section of the **Motor Editing Pane** (Figure 65).



The screenshot shows the 'Position' section of the Motor Editing Pane. It includes a 'Reset Position' button at the top. Below it are several fields: 'Max Fwd Position' (1), 'Min Rev Position' (0), 'Position Scale' (1, highlighted with a red oval), 'Position Units' (counts), 'Max Position Error' (9,999), an 'Abort On Position Error' checkbox (unchecked), and 'Position Error Loaded' (False).

Figure 65

2. Click the **Reset Position** button (Figure 66).



The screenshot shows the 'Position' section of the Motor Editing Pane. The 'Reset Position' button is highlighted with a red oval. The other fields are the same as in Figure 65: 'Max Fwd Position' (1), 'Min Rev Position' (0), 'Position Scale' (1), 'Position Units' (counts), 'Max Position Error' (9,999), 'Abort On Position Error' (unchecked), and 'Position Error Loaded' (False).

Figure 66

3. Enter “0” in the **New Position** field and click **OK** (Figure 67).

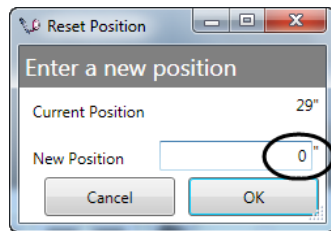


Figure 67

4. In the **Manual Controls** section of the **Motor Editing Pane** (Figure 68) run the motor forward or reverse as far as you safely can.

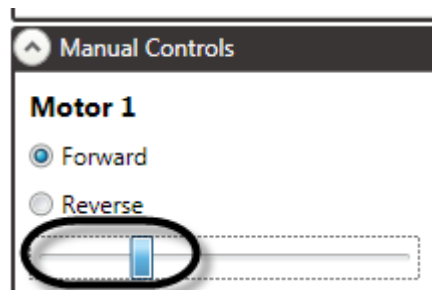


Figure 68

5. At the top of the **Motor Editing Pane**, note the number of encoder counts that are traveled.
6. Measure the physical distance traveled by the scenery.
7. To compute the scaling factor, divide the number of encoder counts traveled by the physical distance.
For example: 1,245,012 encoder counts / 125.5 inches = 9,920.41 encoder counts per inch.
8. Enter the result from the previous step into the **Position Scale** field and then select the appropriate units from the **Position Units** drop-down menu (Figure 69).
In this example, you would enter “9,920.41” and select “inches.”

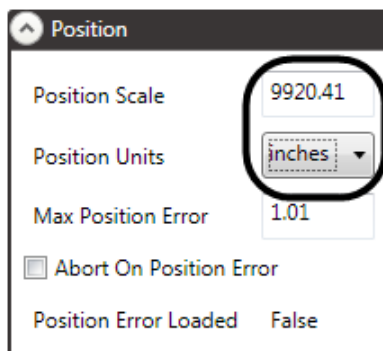


Figure 69

Setting Motor Software Limits: The Min Reverse and Max Forward Positions

All motors in SpikeMark *must* have a **Max Forward Position** and a **Min Reverse Position** defined in the **Position** section of the **Motor Editing Pane**.

NOTE: SpikeMark calculates the track length for all winches in the Stage Model Viewer based on these configuration parameters. By setting these values, SpikeMark will prevent you from writing cues that travel beyond the physical limits of the machine. If you enter in a position of 1000" for a cue, but the **Max Forward Position** is set to 100", SpikeMark will reduce the cue target position to 100" to keep it within the defined limits.

1. Select the motor in the **Cue Grid**.
2. To set the **Min Reverse Position**, go to the **Manual Controls** section in the **Motor Editing Pane** and run the motor in reverse as far as its physical limit will allow. Observe the **Position** value at the top of the **Motor Editing Pane** (circled in Figure 70).

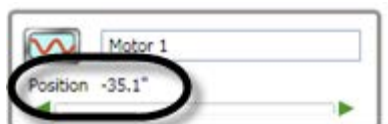


Figure 70

3. Type the **Position** value into **Min Reverse Position** field (Figure 71).

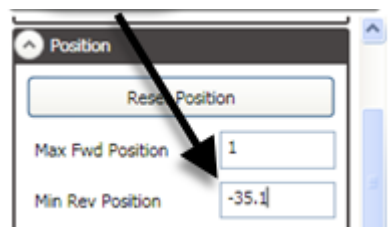


Figure 71

4. To set the **Max Fwd Position**, go to the **Manual Controls** section of the **Motor Editing Pane** and run the motor forward as far as its physical limit will allow. Observe the **Position** value at the top of the **Motor Editing Pane** (circled in Figure 72).



Figure 72

5. Type the **Position** value into **Max Fwd Position** field (Figure 73).

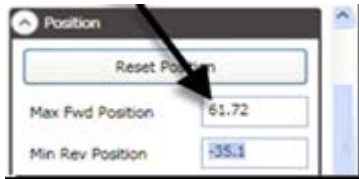


Figure 73

NOTE: The following rules must be followed for **Max Forward Position** and **Min Reverse Position**:

The Max Forward Position must be greater than **Min Reverse Position**.

Values can be negative, provided that the above rule is followed.

Turntables must have software limits, even if there is no physical limitation to how far they can travel. For turntables, you can often just set a high value in both directions [For example: 7200 degrees for the **Max Forward Position** and -7200 degrees for **Min Reverse Position** would allow the turntable to spin around in either direction 20 times].

If possible, set the **Max Forward Position** and the **Min Reverse Position** to be slightly less than the physical limits set on your motor switches (Figure 74).

Physical limits *usually* do not apply to turntables.

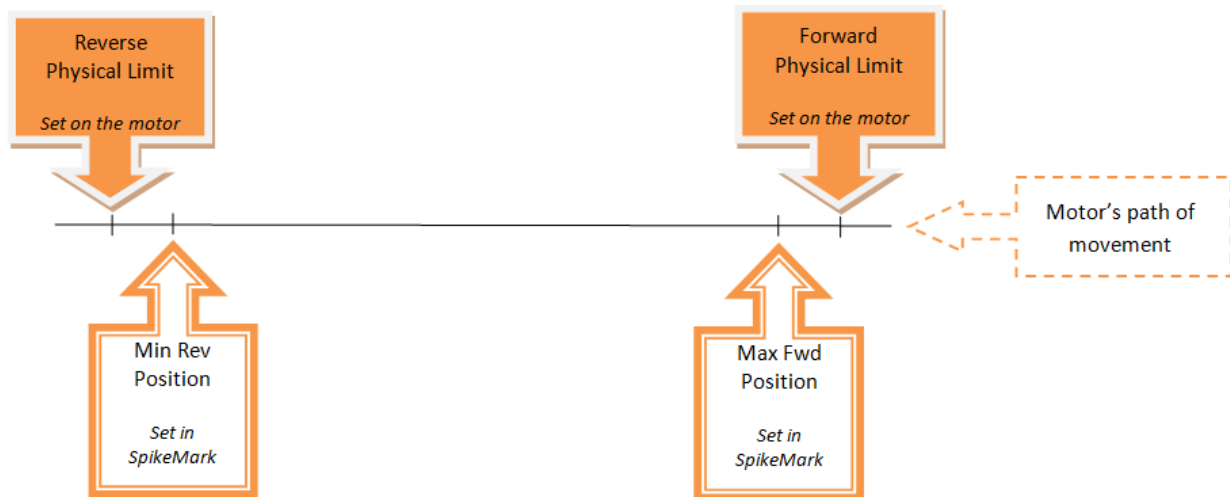


Figure 74

Resetting the Motor's Position Value

Motor position is stored in the Stagehand. SpikeMark simply displays the position information that it receives from the Stagehand. There are times when you need to change the position information that is stored in the Stagehand. For instance, if the Stagehand loses power or is replaced then the position information will be wrong. In such a case, you should move the motor to a known position with the **Manual Controls** on the stage and then reset the position value stored inside the Stagehand.

NOTE: If your production **powers down** every evening, you will need to reset every motor's position value before every show.

1. Select the Motor in the **Cue Grid** and confirm that it is connected to the network (see "Setting Up a Network Connection," page 19).
2. Click the **Reset Position** button in the **Position** section of the **Motor Editing Pane** (Figure 75).



Figure 75

3. In the **New Position** field, enter the *correct Position* value and click **OK** (Figure 76).

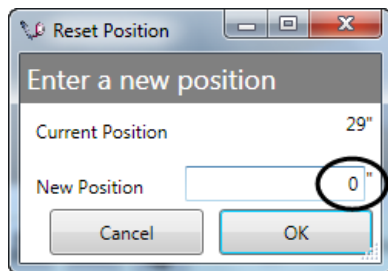


Figure 76

Motor Tuning

Motor tuning is perhaps the most difficult aspect of scenic automation. By tuning a motor, you are describing to a Stagehand how it should regulate the motor speed during a cue using mathematical formulae. We won't get too deep into the intricacies of that math, but it is hard to discuss motor tuning without getting a little involved in the nuts and bolts of motion control theory.

First, it helps to understand the problem that motor tuning solves. The best analogy for motor tuning is cruise control on a car. When you set your cruise control to 55 mph, the car has to adjust the amount of fuel going to the engine to keep moving at that speed. If you are going up hill, the car lags for a second and then more gas rushes into the engine and the car speeds up to get back to the desired speed.

Motion control for scenery is very similar. We program the Stagehands with a speed and target position, and the Stagehand has to adjust power to the motor to achieve the desired effect. If there's a bump in the floor, the winch has to have a little more power to maintain speed. If the winch has too much power, it will lurch over the bump and then quickly reduce power so that it doesn't go too fast once it is over the bump.

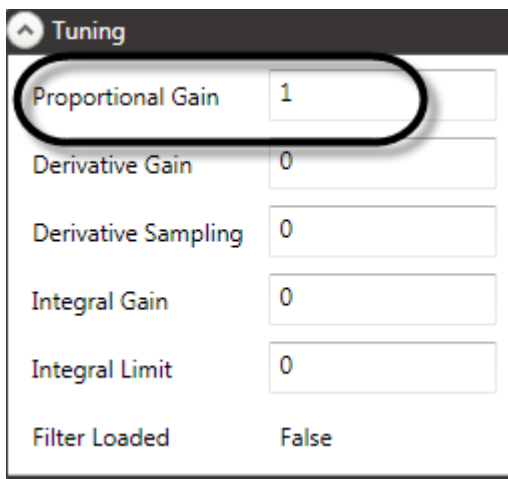
The Stagehand is constantly analyzing where the motor is versus where it should be and then adjusting motor power to minimize the difference between where the motor should be and where it really is. It does this analysis a few million times per second. When it wants to apply power to correct for error in position, it looks to us for guidance.

By entering in some tuning parameters, we are giving the Stagehand that guidance. In a confounding abstract way, we are specifying how much power to give the motor when it needs to make a correction. If the values that we enter give the motor too much power during correction, the motor will be jerky as it over-corrects and then has to pull back (remember, this happens millions of times per second). If the values we enter do not provide enough power to the motor to correct position, it will never reach the cue position since it will run out of power and be unable to muscle the load onto the target.

There are five tuning values that can be used to achieve smooth, accurate motion. To adjust these parameters, select the motor you wish to tune in the **Cue Grid** and select the **Tuning** section of in the **Motor Editing Pane** (Figure 77 through Figure 82).

Proportional Gain

Proportional Gain (Figure 77) is the ratio of power to position error. The higher this value, the more power will be applied to correct for position errors. This is often the only tuning parameter you will need to adjust. **Proportional Gain** needs to be at least “1” for the motor to move at all. If the value is too high, the motor will begin lurching. Sometimes this lurching can be extreme so make sure to have your hand on the Emergency Stop when you first adjust this parameter.



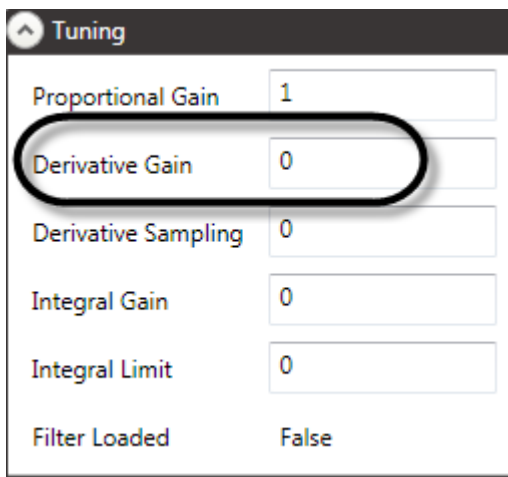
The screenshot shows a 'Tuning' panel with a list of parameters. The 'Proportional Gain' field, which has the value '1', is highlighted with a black rounded rectangle. Below it are 'Derivative Gain' (0), 'Derivative Sampling' (0), 'Integral Gain' (0), 'Integral Limit' (0), and 'Filter Loaded' (False).

Tuning	
Proportional Gain	1
Derivative Gain	0
Derivative Sampling	0
Integral Gain	0
Integral Limit	0
Filter Loaded	False

Figure 77

Derivative Gain

Derivative Gain (Figure 78) is applied against changes in position error. It can have a dampening effect on a motor, absorbing the shocks from a high **Proportional Gain**.



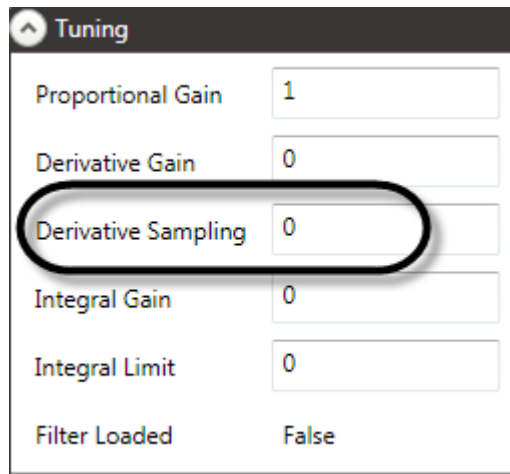
The screenshot shows the same 'Tuning' panel. In this view, the 'Derivative Gain' field, which has the value '0', is highlighted with a black rounded rectangle. The other parameters remain the same as in Figure 77.

Tuning	
Proportional Gain	1
Derivative Gain	0
Derivative Sampling	0
Integral Gain	0
Integral Limit	0
Filter Loaded	False

Figure 78

Derivative Sampling

Derivative Sampling (Figure 79) describes how often the **Derivative Gain** should be applied. If you are using **Derivative Gain**, **Derivative Sampling** must be set to at least “1.” Rarely is there any benefit to using a **Derivative Sampling** value higher than “2.”



The screenshot shows a 'Tuning' panel with the following settings:

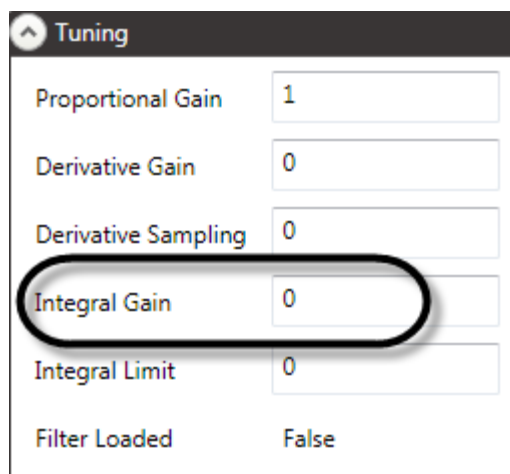
Parameter	Value
Proportional Gain	1
Derivative Gain	0
Derivative Sampling	0
Integral Gain	0
Integral Limit	0
Filter Loaded	False

The 'Derivative Sampling' field is highlighted with a black rounded rectangle.

Figure 79

Integral Gain

Integral Gain (Figure 80) tries to compute corrective action by evaluating the position error and motor response over time. From a practical standpoint, **Integral Gain** will often correct the motor position at the end of travel. This can be helpful if you need to reduce **Proportional Gain** to smooth out a motion, but then lose accuracy. In such a scenario, adding in 1 or 2 points of **Integral Gain** will often retain accuracy without sacrificing smooth travel.



The screenshot shows a 'Tuning' panel with the following settings:

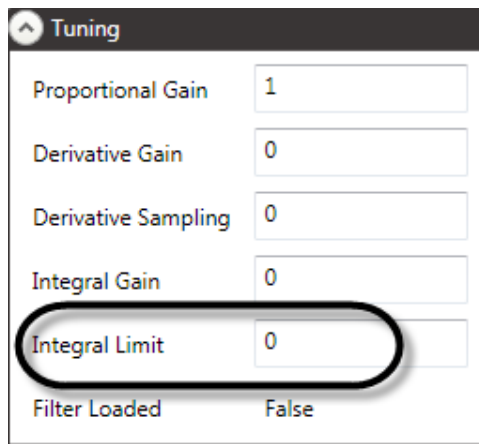
Parameter	Value
Proportional Gain	1
Derivative Gain	0
Derivative Sampling	0
Integral Gain	0
Integral Limit	0
Filter Loaded	False

The 'Integral Gain' field is highlighted with a black rounded rectangle.

Figure 80

Integral Limit

Integral Limit (Figure 81) prevents a condition known as Integral Windup, which can lead to really erratic motor behavior. If you use an **Integral Gain**, set **Integral Limit** to 1000.



The screenshot shows a 'Tuning' window with the following parameters:

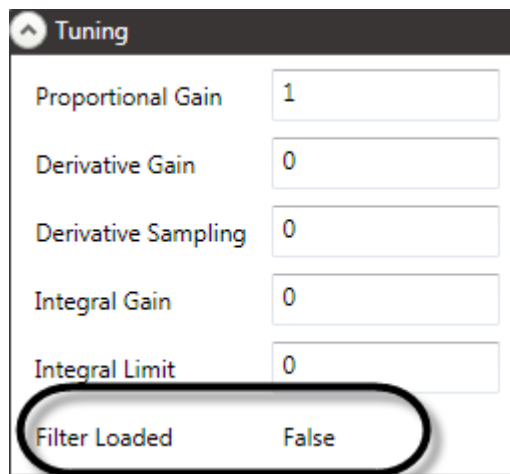
Parameter	Value
Proportional Gain	1
Derivative Gain	0
Derivative Sampling	0
Integral Gain	0
Integral Limit	0
Filter Loaded	False

The 'Integral Limit' field is highlighted with a red oval.

Figure 81

Filter Loaded

The **Filter Loaded** field (Figure 82) is *not* an adjustable motor tuning parameter. Rather, it indicates whether the filter parameters have been sent to the motor yet. After you change a tuning parameter, that change is saved until the next cue is run. When the next cue is run, the new filter parameters will be sent to the motor automatically.



The screenshot shows the same 'Tuning' window as Figure 81, but with the 'Filter Loaded' field highlighted with a red oval.

Parameter	Value
Proportional Gain	1
Derivative Gain	0
Derivative Sampling	0
Integral Gain	0
Integral Limit	0
Filter Loaded	False

Figure 82

Over the past several years, we've come up with a practical approach to motor tuning that yields good results in a reasonable amount of time. That approach is shown in the form of a flow chart (Figure 83). The steps are clear and simple, but it can still be a frustrating experience the first time you attempt to tune a motor. It's extremely important to find at least an hour of quiet time on the stage without anyone else working around the moving scenery. Get a cup of coffee, relax, focus and take a stab at it. If that doesn't work, give us a call (see the "Contact Us" section on page 157). We have had plenty of experience helping people tune motors over the phone. The good news is that it gets a lot easier once you see it work the first time.

Motor Tuning Flow Chart

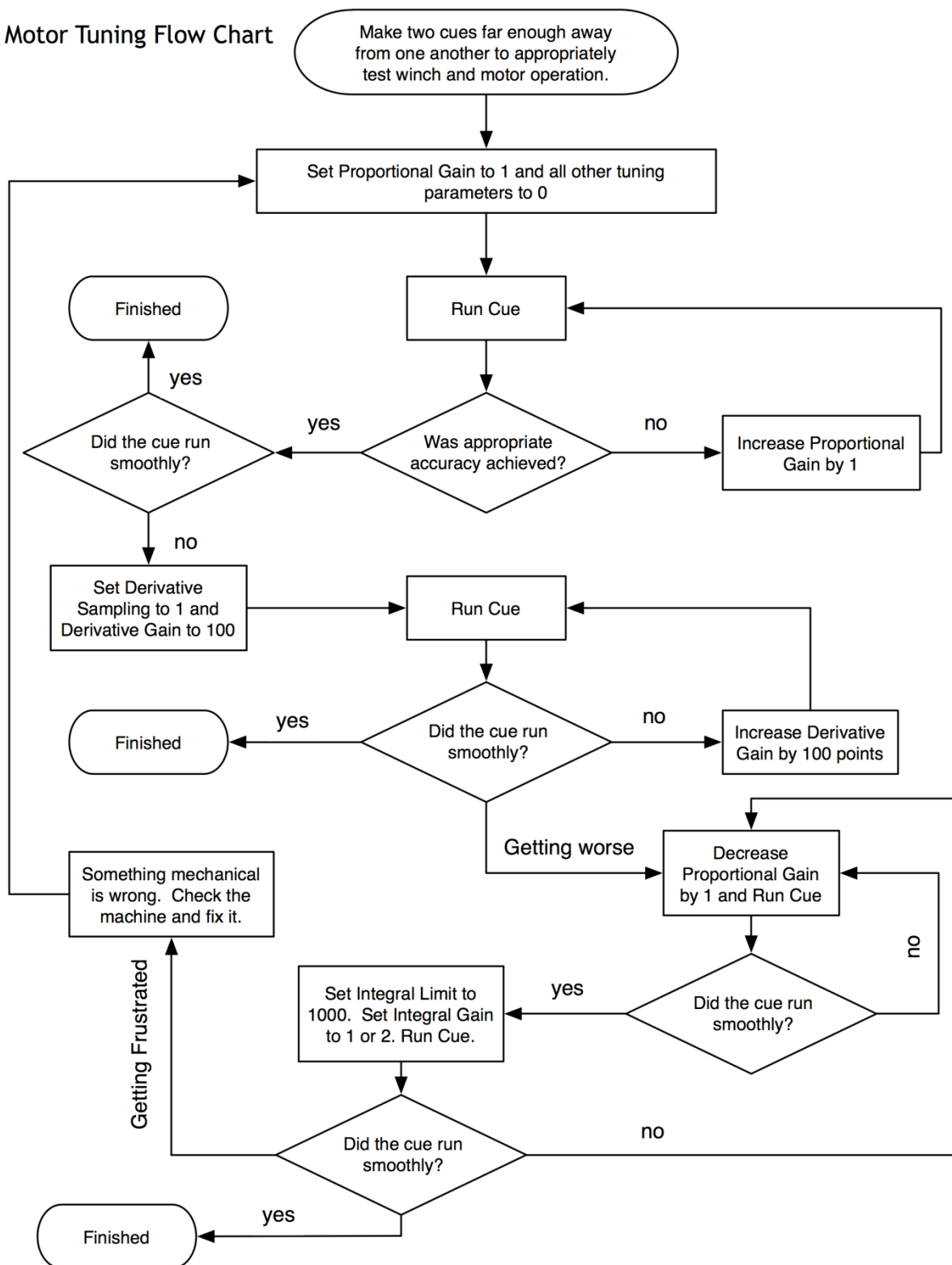


Figure 83

Setting the Max Position Error

Once a motor has been adequately tuned, it is a good idea to set a **Max Position Error** value and make sure the **Abort On Position Error** field is selected in SpikeMark. This protects you from runaway motors when the encoder is damaged or disconnected.

Without a **Max Position Error**, the Stagehand will send an increasing amount of power to compensate for a position error. If the encoder is not sending information to the Stagehand, the Stagehand will think that the motor is not moving. So it will rapidly work up to full speed and run at full speed indefinitely, since, from the Stagehand's perspective, the motor is standing still. Without setting **Max Position Error**, you will be forced to use the Emergency Stop (but not before you have been scared out of your skin).

A **Max Position Error** value can also stop a motor when it is poorly tuned, but not posing any danger. This interrupts the show and can be detrimental to the performance. To keep the motor from faulting during normal circumstances, the default value for maximum position error is set as high as possible. You can tighten the allowable position error by lowering this value if you have requirements for a more exacting motion profile.

NOTE: The maximum allowable **Max Position Error [MPE]** value is 9,999 raw encoder counts after scaling.

The **MPE** is given in scaled position units such as inches or degrees

Therefore, if you have a position scale of 10,000 encoder counts per inch, your MPE will equal 1" $[9,999 \text{ raw encoder counts} \div 10,000 \text{ encoder counts/inch} = 1 \text{ inch}]$.

If you then manually increased the MPE to 60, this would increase the raw encoder count value to 599,940, which would exceed the 9,999 raw encoder count maximum value.

1. Select the motor you wish to adjust in the **Cue Grid**.
2. Check the **Abort On Position Error** option in the **Position** section of the **Motor Editing Pane** (Figure 84). If you like, you can tweak the value of the **Max Position Error** (this value is given in scaled position units such as inches or degrees, etc.).

Figure 84

NOTE: Position error changes are loaded into the Stagehand prior to running a cue.

Setting the Motor Cue Completion Mode

SpikeMark has two modes for running motors:

Target Tolerance

Target Tolerance mode is one of SpikeMark's newest features and has been implemented to make motor cues more reliable regardless of poor tuning. When using **Target Tolerance** mode, the motor will continue to try to achieve target regardless of how much time has passed. Once the motor position is within your specified tolerance, it will turn off and display a **Complete** status icon.

NOTE: **Target Tolerance** is the default mode in SpikeMark.

Strict Timing

Strict Timing mode insists that the motor move accurately and achieve the cue position within the theoretical time allowed for that movement based on the programmed distance, speed, and acceleration. If the motor cannot achieve target in the computed time, it will be turned off regardless of its current position.

NOTE: For shows that have tight timing requirements, this option may be useful. However, it has mostly been included in SpikeMark since it was the only mode available in earlier versions of SpikeMark. When using this mode, you have to be diligent about tuning your motors well. Sloppy tuning will result in motors abruptly stopping in mid-move.

1. To specify which cue completion mode to use, select the motor you wish to edit in the **Cue Grid**.
2. Scroll to the **Cue Behavior** section in the **Motor Editing Pane** (Figure 85).

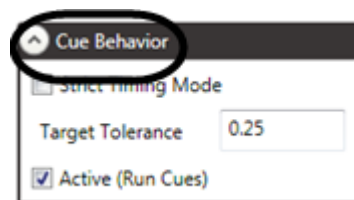


Figure 85

3. To use **Target Tolerance** mode, make sure that **Strict Timing Mode** is *deselected* and enter a **Target Tolerance** value, which is listed in scaled position units (for example, inches, degrees, etc.).
In Figure 86, “0.25” equals 1/4 inch.

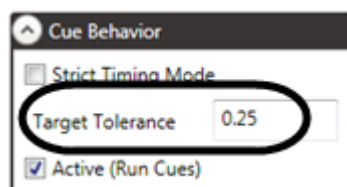


Figure 86

4. To use **Strict Timing Mode**, select the **Strict Timing Mode** field (Figure 87).
With **Strict Timing Mode** checked, the **Target Tolerance** value will be ignored.

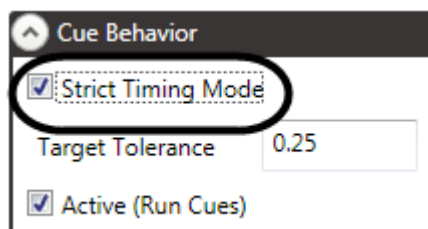


Figure 87

NOTE: Target Tolerance mode is the default mode in SpikeMark.

Setting a Motor's Active Mode

Most of the time, you will want the motor to load and run cues normally. However, there are occasions when a motor needs to be temporarily removed from the show. Perhaps there is a problem with the scenery on a winch track, or perhaps you need to run through movements using only the motors downstage and don't want any upstage motors to move.

To temporarily disable a motor, deselect the **Active** box (Figure 88) in the **Cue Behavior** section of the **Motor Editing Pane**. The motor will not run cues until the **Active** box is checked again.

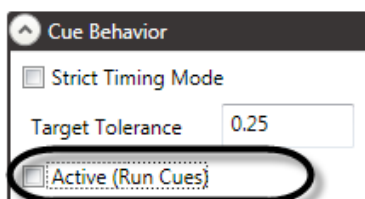


Figure 88

When a motor is deactivated (not **Active**), it will not impede the operation of cue *links* (see “Cue Links” on page 96). Normally a cue needs to successfully load all motor movements before it can be executed by a cue link. However, if a motor within in a cue is *not Active* and thereby unable to move, the entire cue will *still* be allowed to execute. SpikeMark will trust that *you have ascertained that it is safe for the cue link to run without this motor* and you are overriding the normal requirement that all motors must be able to move before any motor will move in a linked-cue sequence.

EXAMPLE:

- 1: Assume you have written Cue 1, in which WAGON A moves upstage **and** a CURTAIN goes up.
- 2: Also assume you have written Cue 2, in which WAGON B moves downstage, past the CURTAIN and meets WAGON A.
- 3: Now assume that Cue 1 and Cue 2 are linked—by position—so that WAGON B starts moving once WAGON A has moved [for this example, 10"].
- 4: Lastly, assume that you want to rehearse WITHOUT the CURTAIN.
- 5: To do this, you would fly the curtain up and make sure it is out of the way of WAGON B. Next, you would inactivate the CURTAIN [by deselecting the **Active** field in the **Cue Behavior** section of the curtain's **Motor Editing Pane**]. Then it would then be safe to test WAGON A and WAGON B, as you've moved the curtain up and out of the way of WAGON B
- 6: Deactivating a motor does NOT stop the links between cues. Therefore, the link between Cue 1 and Cue 2 will still work [in this example, WAGON B will start moving once WAGON A has reached the 10" mark].

Also see *Figure 89*.

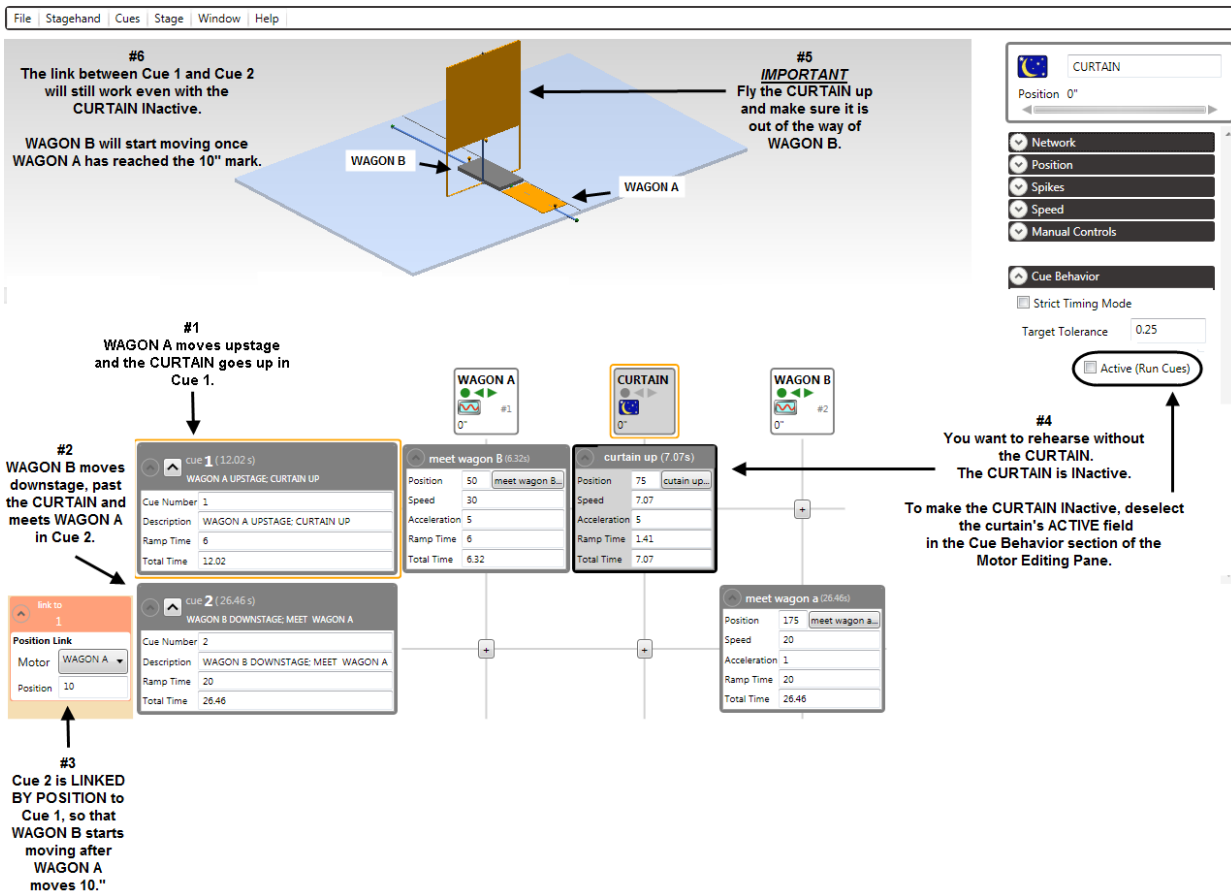


Figure 89

It is worth mentioning here that the **Active** state of a motor can also be controlled through a Stagehand FX input action. Using an input on a Stagehand FX, you can quickly deactivate one or many motors. (To learn more about Stagehand FX input actions, see "The Stagehand FX" section on page 65).

Changing the Status Refresh Rate

Whenever SpikeMark connects to a motor, it starts asking the Stagehand for updated information about its status, position, limits, and Emergency Stop condition. SpikeMark asks for updated information several times a second to give the illusion of a steady stream of information. This is how the position information appears to fluidly change as the motor moves on stage. Most of the time, this all works great and you don't have to think about *how* it works. However, if you have a large show, or a slow computer, you may find that SpikeMark seems to be slowing down or feeling clunky.

In the case described above, you may wish to tweak the frequency of information updates, known as the **Polling Interval**, so that SpikeMark doesn't get bogged down with this constant updating process.

1. Select the motor you wish to adjust in the **Cue Grid** or **Stage Model**.
2. Scroll to the **Advanced** section of the **Motor Editing Pane** (Figure 90).

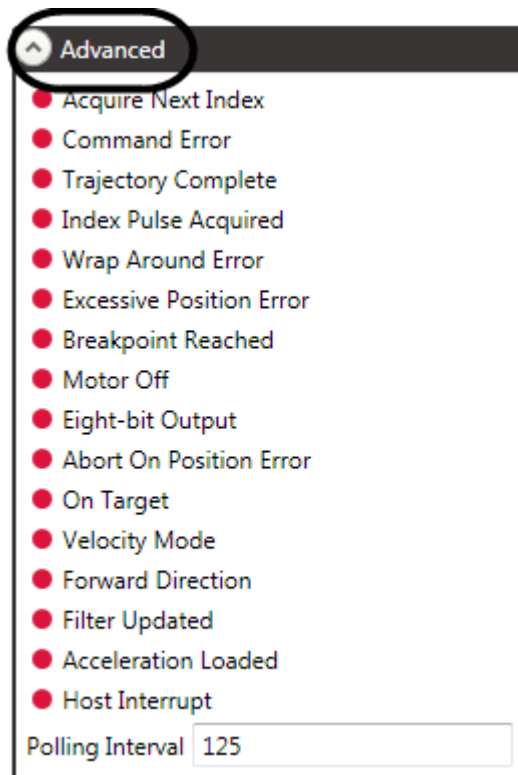


Figure 90

3. Enter a new value in the **Polling Interval** field (Figure 91).
The values are in milliseconds. The default value is “125,” which means that SpikeMark updates motor information 8 times a second. Setting the Polling Interval to “250” will update motor information 4 times a second, which will relieve the computer of some processing stress, but the updates on screen will appear jerky. Setting the polling interval to 50 will provide updates 20 per second making the information updates very smooth and snappy, but eating more of your computer’s resources.

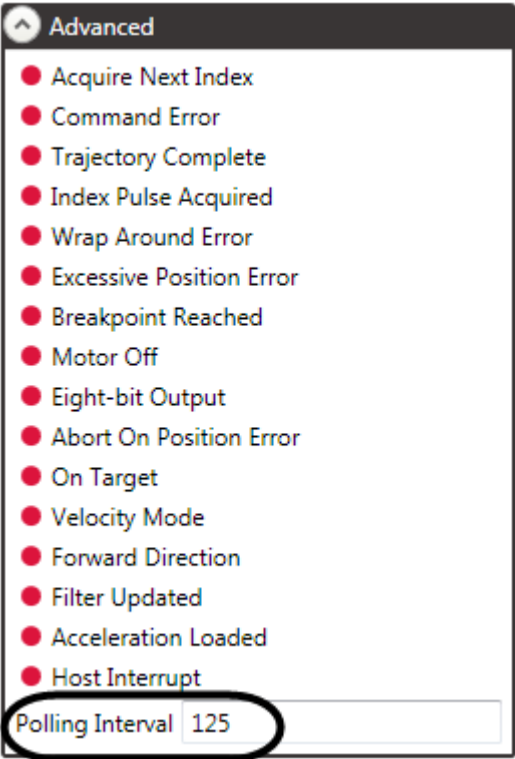


Figure 91

The Stagehand FX

A Stagehand FX (or FX) is an input/output device that has a collection of parameters which are typically set once during a production. It differs from a Stagehand *Motor*, which controls the position of a motor, whereas the Stagehand FX turns any device (motor, valve, etc.) on or off. The **Stagehand FX Editing Pane** is on the right side of the SpikeMark screen and allows you to edit all of the FX configuration parameters for the show (Figure 92).

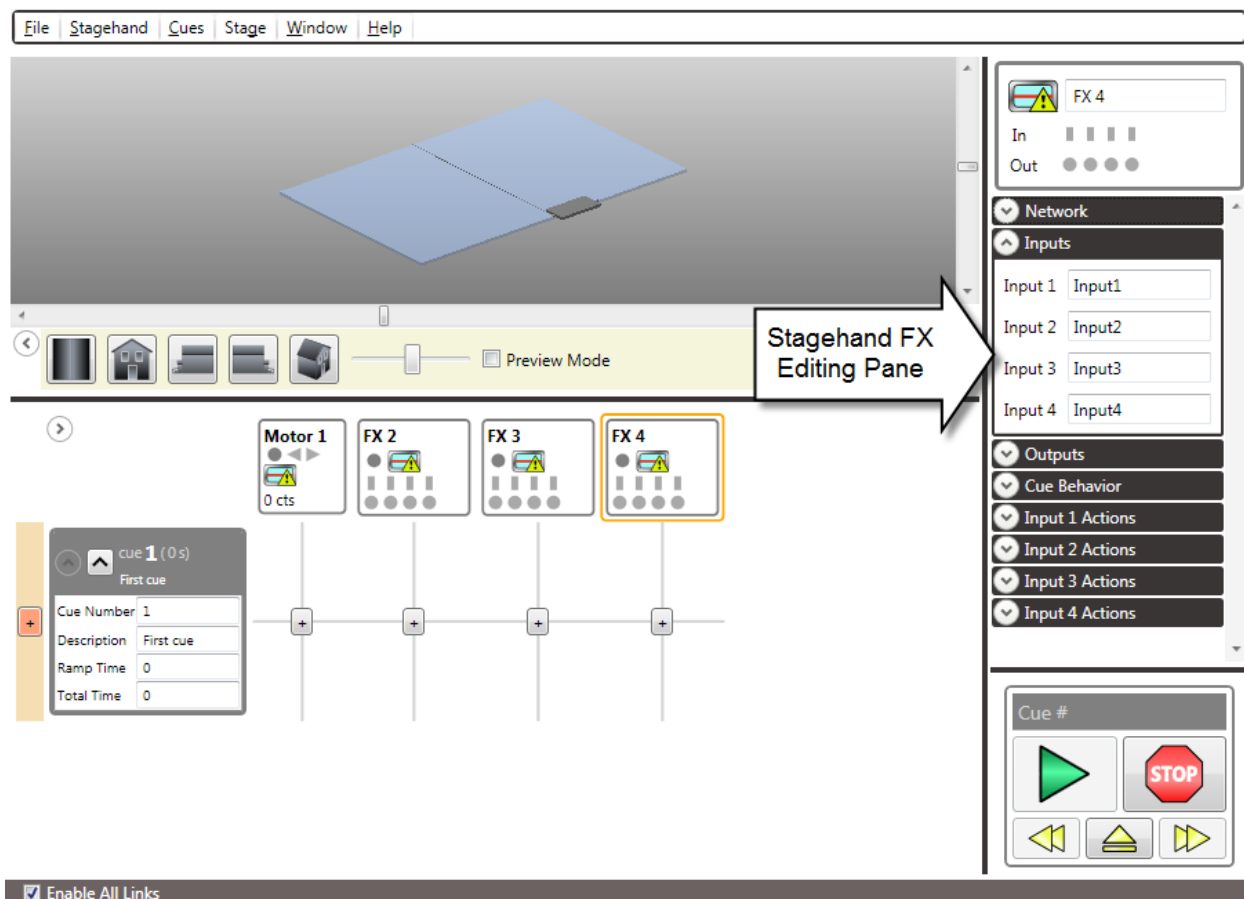


Figure 92

Adding a New Stagehand FX

1. Right-click in the **Cue Grid** and select **Add Stagehand FX** (Figure 93).



Figure 93

Or, in the **Stagehand** menu, click **Add Stagehand FX** (Figure 94).

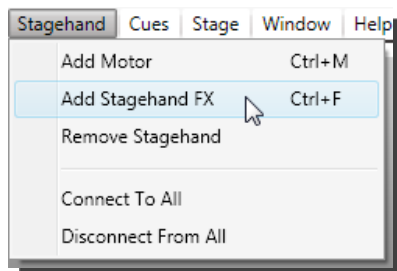


Figure 94

Or, press **Ctrl + F**.

Stagehand FX Configuration

1. Select the Stagehand FX you wish to update in the **Cue Grid**.
The selected Stagehand FX will be indicated with an orange outline (Figure 95).

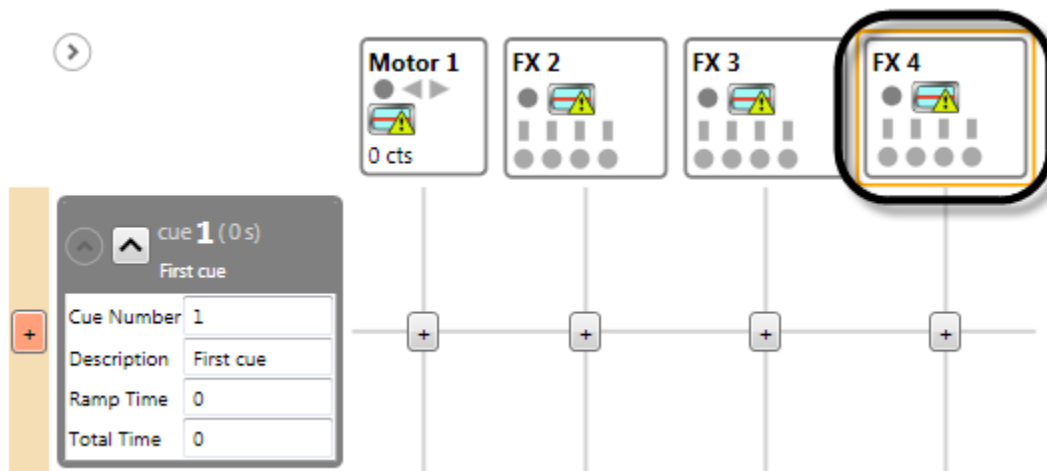


Figure 95

2. Once you have selected an FX in the **Cue Grid**, you can begin editing its parameters in the **FX Editing Pane**, the scrolling window on the right side of the SpikeMark screen that contains sections of parameters grouped by function. You can collapse or expand these sections to show or hide the information (Figure 96).

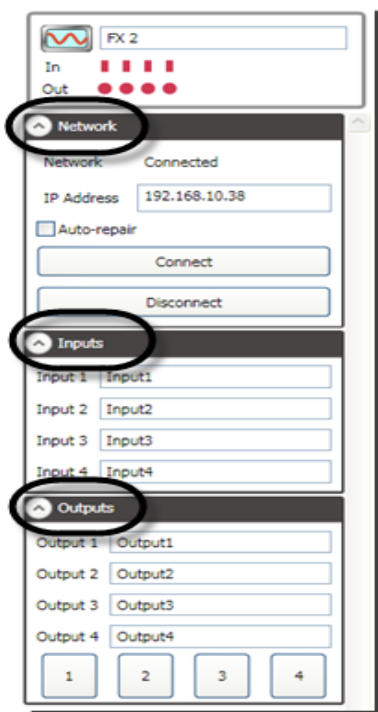


Figure 96

Renaming an FX

To help identify all the Stagehand FX devices in your show, you can give each FX a unique name. Additionally, you can rename all of the Inputs and all of the Outputs for an FX to help remember the purpose of each switch in the Stagehand FX.

1. Select the Stagehand FX to be edited in the **Cue Grid** and then select the field that contains the default name.

In Figure 97, the default name is “FX2.”



Figure 97

2. Type a new name into the field and press **Tab** key or click out of the field to update the name in the **Cue Grid**.

In Figure 98, the new name is “Trap Door.”



Figure 98

Renaming FX Inputs and Outputs

1. Select the Stagehand FX to be edited in the **Cue Grid**
2. In the **Inputs** section of the FX Editing Pane, enter the new **Input** name(s) and in the **Outputs** section of the FX Editing Pane, enter the new **Output** name(s) (Figure 99).

Inputs

Input 1

Lift Limit

Input 2

Input2

Input 3

Input3

Input 4

Input4

Outputs

Output 1

SR Lock

Output 2

SL Lock

Output 3

SR Cylinder

Output 4

SL Cylinder

1

2

3

4

Figure 99

FX Status

Stagehand FX Status information is displayed next to the Stagehand FX name in the **Stagehand FX Editing Pane** and in the **Cue Grid** column headers. The status of the Stagehand FX device is indicated by various icons. Each icon status is described below.

Icon	Status
	FX disconnected from the network.
	FX connected to the network, but idle.
	FX is inactive and won't run cues or affect cue links.
	FX has completed a cue.
	Emergency stop.
	Cue loaded.
	Soft stopping, but not yet fully stopped.
	Stopped mid-cue by a soft-stop command.
	An output is active in manual mode.
	One (or more) output(s) is running in a cue.
	All outputs have been turned off because of a fault condition.

Manually Activating FX Outputs

If you want to test a Stagehand FX output switch, you can use the manual output buttons in the Stagehand FX Editing Pane. Press the button that corresponds to the output you wish to activate. The button behaves like a momentary switch; the FX output will be active as long as the button is held down. When you release the button, the FX output will deactivate.

1. Select the Stagehand FX you wish to activate in the **Cue Grid**.
2. Click and hold down one of the output buttons in the **Outputs** section of the **FX Editing Pane** (Figure 100).

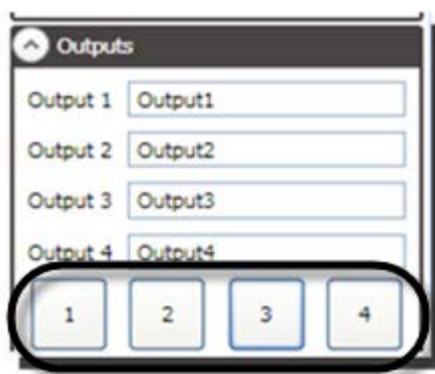


Figure 100

3. Release the button to deactivate the output.

Setting the FX Active Mode

Most of the time, you will want the Stagehand FX to load and run cues normally. However, there are occasions when an effect needs to be temporarily removed from the show. Perhaps there is a problem with the scenery on a winch track, or perhaps you need to run through movements using only the Stagehands downstage and don't want anything upstage to move.

1. Select the Stagehand FX you wish to disable temporarily in the **Cue Grid**.
2. In the **Cue Behavior** section of the **Stagehand FX Editing Pane**, deselect the **Active** field (Figure 101).
*The FX will not run cues until the **Active** field is selected again.*

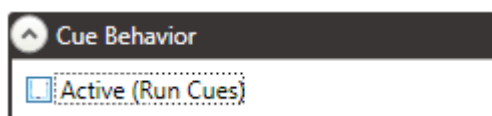


Figure 101

NOTE: When a Stagehand FX is deactivated (not **Active**), it will not affect the operation of cue *links* (see “Cue Links” on page 96). Normally a cue needs to successfully load all motor and FX movements before it can be executed by a cue link. However, if a Stagehand FX within in a cue is *not Active*—and thereby unable to turn a motor on or off—the entire cue will *still* be allowed to execute. SpikeMark will trust that *you have ascertained that it is safe for the cue link to run without this FX* and that you are overriding the normal requirement that all FXs must be able to turn a motor on or off before a linked-cue sequence initiates.

EXAMPLE: Assume you have written Cue 1, in which a LIGHT [controlled by a Stagehand FX] goes on.

Also assume you have written Cue 2, in which a TAB CURTAIN [controlled by a Stagehand FX] opens and WAGON A [controlled by a Stagehand Motor] moves down stage from behind the curtain to in front of the curtain.

Now assume that Cue 1 and Cue 2 are linked—by position—so that the LIGHT goes ON when WAGON A travels to position 60”.

Lastly, assume that you want to rehearse WITHOUT the TAB CURTAIN.

To do this, you would open the TAB CURTAIN, to make sure it is out of the way of WAGON A. Next, you would deactivate the TAB CURTAIN [by deselecting the **Active** field in the Cue Behavior section of the tab curtain’s FX Editing Pane]. Then, it would then be safe to test MOTOR A and the LIGHT, as you’ve moved the curtain up and out of the way of MOTOR A.

Deactivating the FX does not stop the links between the cues. The link between Cue 1 and Cue 2 will still work [in this example, the LIGHT will go ON once MOTOR A has reached the 60” mark].

The **Active** state of a Stagehand FX can be controlled through any FX input action. Using an input on a Stagehand FX, you can quickly deactivate one or many Stagehand Motors or FXs. See the “FX Input Action” section on page 73 for more information.

FX Input Action

Stagehand FX Inputs can be used in a couple of different ways to help create complex cue sequences. In the “Linking Cues by FX Input” section on page 101, you’ll learn how to use FX Inputs to fire cue links, but here we will illustrate another way to use FX Inputs: Input Actions.

Input Actions introduce a way to trigger events in the show that are not directly related to a cue. FX Input Actions give you a way to tell SpikeMark to do something whenever an input is activated, regardless of what cue is running. We are starting modestly with two action types, but look for more actions in future releases.

Deactivating Stagehand FX Action

You can define an Input Action that will toggle the **Active** state of one or more Stagehand Motors *or* FXs to temporarily keep them from running cues during the show. This may be handy if you have a scenic element that is involved in a number of cues, but requires tight synchronization with an actor’s movements. Some nights the computer sequencing works fine, but some nights you need to take manual control of the motor and jog it to match the actor. When that happens, you don’t want to mess up all the other cues in the sequence, so you wire an “Override” button into a Stagehand FX and configure it as illustrated in the steps below.

1. Select the Stagehand FX you wish to configure in the **Cue Grid**.
2. In the **Input (1, 2, 3 or 4) Action** section of the **FX Editing Pane**, click the **Add Action** button (Figure 102).

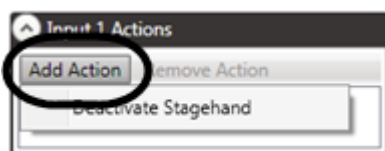


Figure 102

3. Select **Deactivate Stagehand** as the action type (Figure 103).



Figure 103

4. From the **Deactivate Stagehand** drop down menu, select the FX or Motor that should be deactivated when the FX Input is closed (Figure 104).

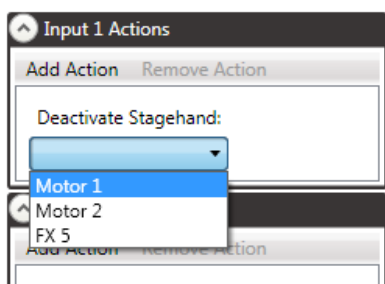


Figure 104

5. Repeat as necessary if you would like a single FX Input to deactivate multiple Stagehands—Motors or FXs (Figure 105).

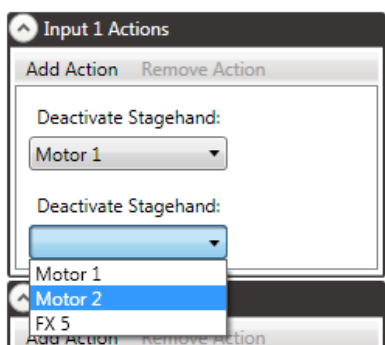


Figure 105

Removing a Stagehand FX Input Action

1. Select the Stagehand FX for which you wish to remove the Input Action in the **Cue Grid**.
2. In the **Input (1, 2, 3 or 4) Action** section of the **FX Editing Pane**, highlight the desired action and press **Remove Action** (Figure 106).

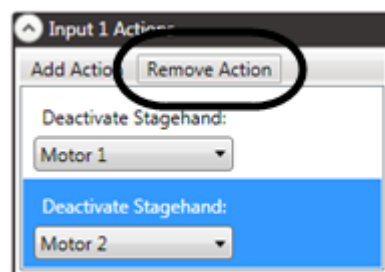


Figure 106

Soft Stopping FX Action

There are times when you may need to soft stop a device in a show based on actor timing. For example, maybe you have two concentric turntables, which both start moving at the same time, but one which needs to stop when an actor says a particular line. In this example, you would not

want to set up a cue in which the turntable reached a target position, but, instead, you could use an FX input and set it so that when manually activated after the actor spoke the line, the motor would slow down to its programmed deceleration rate and then stop.

1. Select the Stagehand FX you wish to configure in the **Cue Grid**.
2. In the **Input (1, 2, 3 or 4) Action** section of the **FX Editing Pane**, click the **Add Action** button (Figure 107).

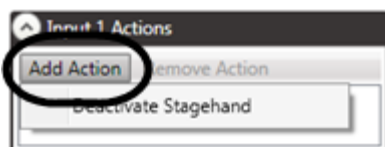


Figure 107

3. Select **Soft Stop Stagehand** as the action type (Figure 108)

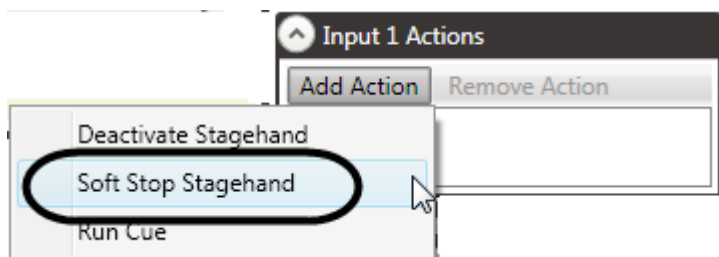


Figure 108

4. From the **Soft Stop Stagehand** drop down menu, select the FX or Motor that should be soft stopped when the FX Input is closed (Figure 109).

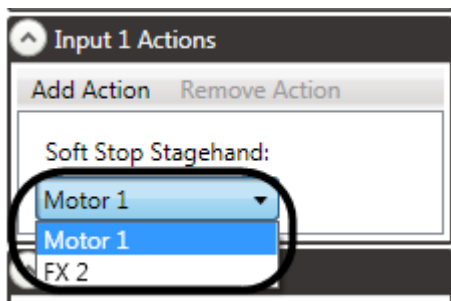


Figure 109

5. Repeat as necessary if you would like a single FX Input to deactivate multiple Stagehands—Motors or FXs (Figure 110).

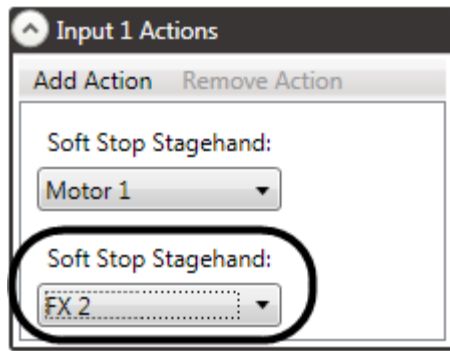


Figure 110

Network Connections

Editing Stagehand Network Settings

The **Network** section of the **Stagehand Motor** or **FX Editing Pane** (Figure 111) contains the settings that pertain to the Ethernet connection between SpikeMark and the Stagehand Motor or FX. The fields in the **Network** section are described below:

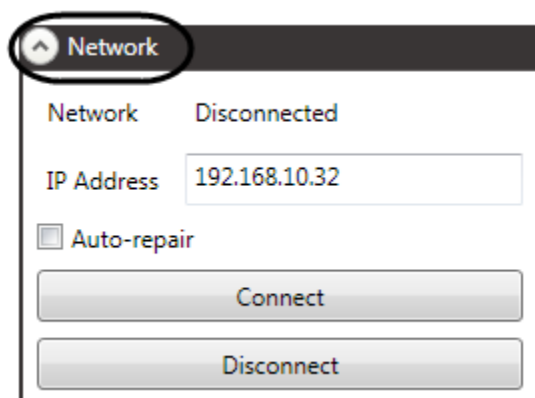


Figure 111

The **Network** field (Figure 112) shows the current status of the network connection. The possible status values are:

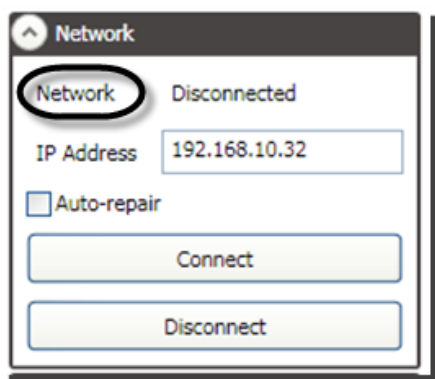


Figure 112

Disconnected: The network connection has not been made yet.

Connecting: A connection is trying to be established.

Errors: The connection failed.

Connections can fail for various reasons, but commonly: a cable was removed, the specified Stagehand address is wrong, or the computer running SpikeMark has an incompatible

address (see the “Setting Up a Network Connection” section on page 19 for steps to configure your computer’s IP address).

The **IP Address** field (Figure 113) contains the IP Address of the selected Stagehand Motor or FX that you wish to connect to. Enter the **IP Address** exactly how it is displayed on the Stagehand Motor or FX face panel. Valid address range from 0.0.0.0 to 255.255.255.255, but remember that the first three segments of the address must match between the computer running SpikeMark and all the Stagehands on the network, while the last segment must be unique for each device.

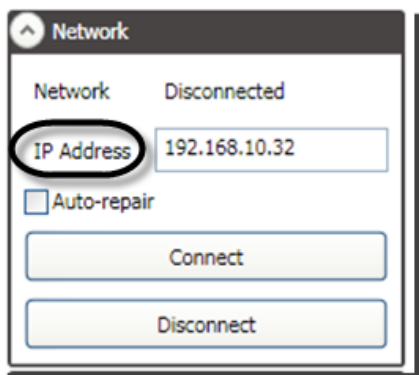


Figure 113

If the **Auto-repair** (Figure 114) field is selected, SpikeMark will automatically try to reconnect to a Stagehand Motor or FX if the connection is lost for any reason. Most of the time you’ll turn this option on for running shows, but you may want to turn it off during initial set up to take a motor ‘offline’.

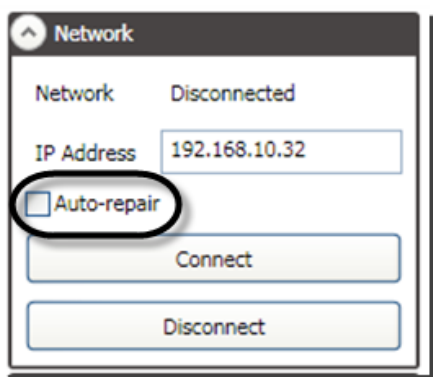


Figure 114

Clicking the **Connect** button (Figure 115) will begin connecting to the Stagehand Motor or FX.

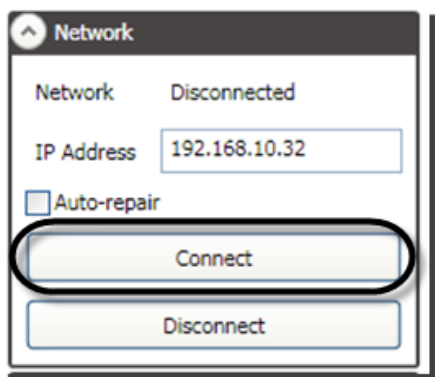


Figure 115

Click the **Disconnect** button (Figure 116) to disconnect from the Stagehand Motor or FX.

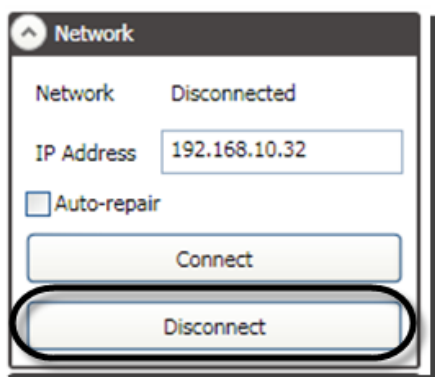


Figure 116

Connecting to a Stagehand Motor or FX

All communications between SpikeMark and different Stagehand Motors or FXs happen over the network. To run any cues, move any motors, or activate an FX output, a network connection must be made to the Stagehand Motor or FX you wish to control. The connection only needs to be established once per show, unless the network connection is lost unexpectedly. There are three ways to establish a network connection.

1. Select the motor or FX you wish to connect to from the **Cue Grid**.
2. In the **Network** section of the **Motor Editing Pane**, click the **Connect** button (Figure 117).

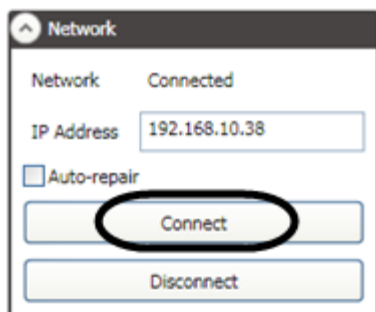


Figure 117

Or, you can connect to all the Stagehand Motors and FXs in the show at once by clicking **Connect To All** in the **Stagehand** menu (Figure 118).

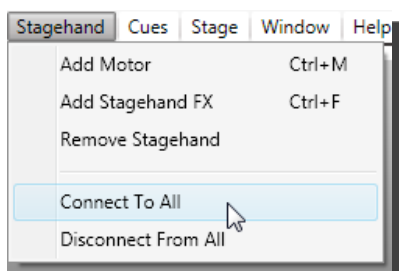


Figure 118

Or, each time you open a show file, SpikeMark will automatically attempt to connect to all the Stagehand Motors and FXs in the show.

Disconnecting from a Stagehand Motor or FX

You can break the network connection to a Stagehand Motor or FX any time you need to temporarily remove it from SpikeMark's control. Disconnecting from the Stagehand Motor or FX will turn off all position & status updates from the Motor or FX. If you would like to *temporarily* remove the Motor or FX from the show, but maintain network communication, see the "Setting a Motor's Active Mode" section on page 60 or the "Setting the FX Active Mode" on page 71.

NOTE: Make sure you deselect the **Auto-repair** field in the **Network** section of the **Stagehand Motor** or **FX Editing Pane** if you want to disconnect from a motor or FX or SpikeMark will immediately try to reconnect.

There are two ways to break a network connection.

1. Select the motor or FX you wish to disconnect in the **Cue Grid**.

2. Press the **Disconnect** button in the **Network** section of the **Stagehand Motor** or **FX Editing Pane** (Figure 119).

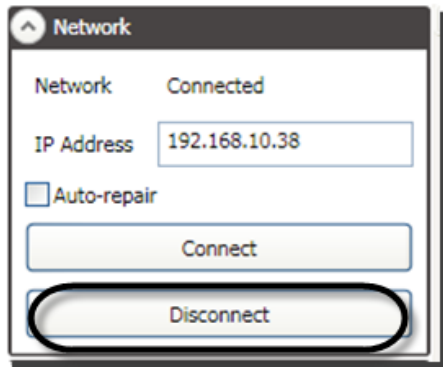


Figure 119

Or, to disconnect from *all* Stagehand Motors or FXs in the show, in the **Stagehand** menu, click **Disconnect From All** (Figure 120).

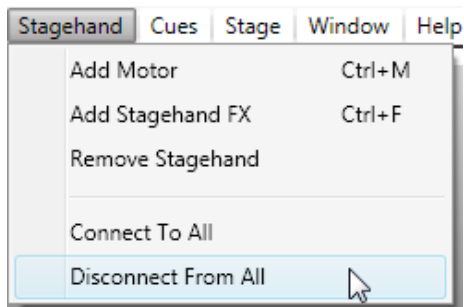


Figure 120

Writing Cues

A Cue is a collection of movements including any number of motors or FX outputs that all start when you press the **GO** button on the **Showstopper** or on the **Cue Controller**. You can have an unlimited number of cues to your show. Each cue has a cue number that can be any decimal number, allowing for ‘point cues’ (for example: 1.5, 209.52, etc.). SpikeMark displays all the cues in the show in the **Cue Grid** as rows sorted by Cue Number. When **SpikeMark** is first launched, a single cue, Cue #1, is created by default (Figure 121). You can use this cue as a starting point or delete it and start with a blank slate.

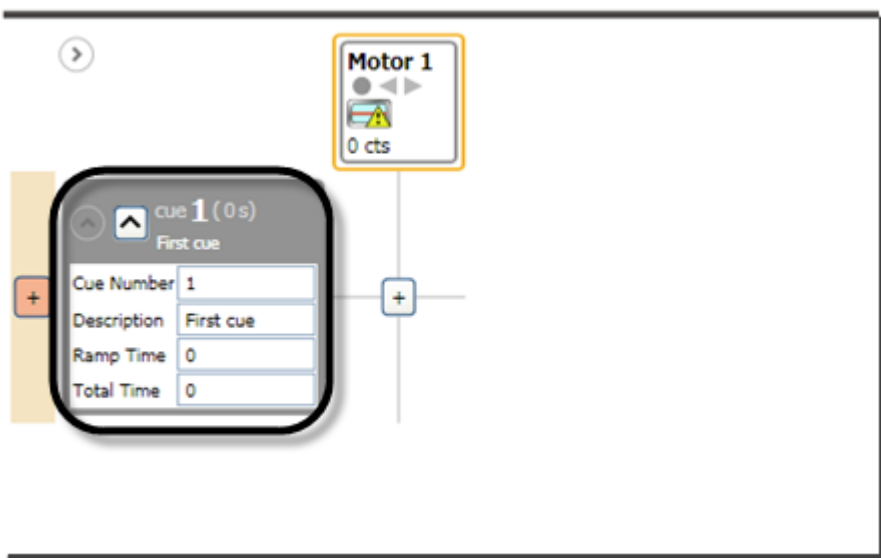


Figure 121

NOTE: You can write cues with or without being connected to any Stagehand Motors or FXs. You can “rough in” all of your cues before setting up your motors on stage. By using the **Spikes** feature and a scaled ground plan, you can write all of the cues for the show and then tweak final positions when the scenery is installed (see the “Spikes” section on 125).

Adding Cues

There are three ways to add cues to your show.

1. Right-click on a blank area in the **Cue Grid** and select **Add Cue** (Figure 122).

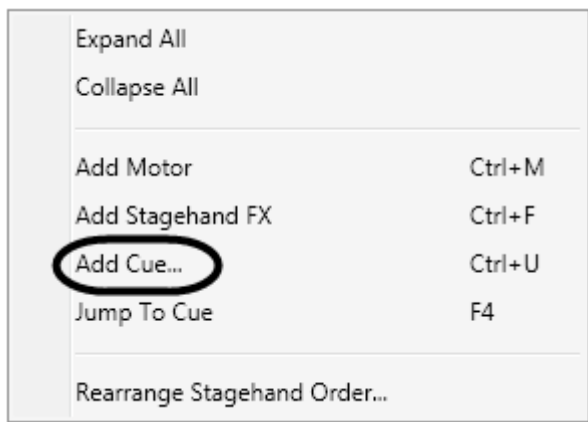


Figure 122

Or, in the **Cues** menu, click **Add Cue** (Figure 123).

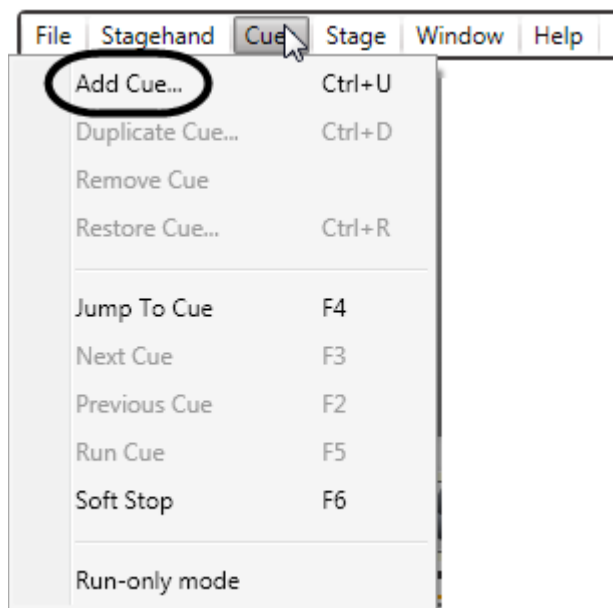


Figure 123

Or, use the keyboard shortcut **Ctrl + U**.

2. A dialog appears prompting you for a **Cue Number** and **Cue Description** (Figure 124).

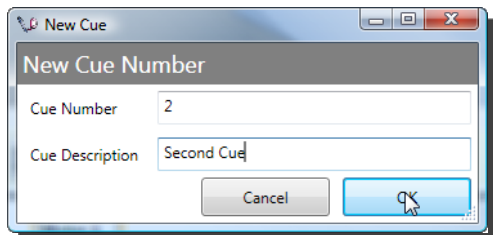


Figure 124

NOTE: You can edit the **Cue Number** or accept the default. The **Cue Description** is solely for your benefit. You can enter a descriptive name to help identify the cue in the **Cue Grid**.

3. The newly created cue will appear in the **Cue Grid** (Figure 125).

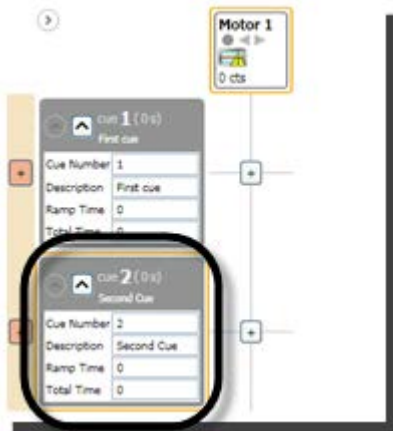


Figure 125

Adding Motor Movements to a Cue

1. Press the “+” button that is displayed at the intersection of the cue row and the motor column (Figure 126).

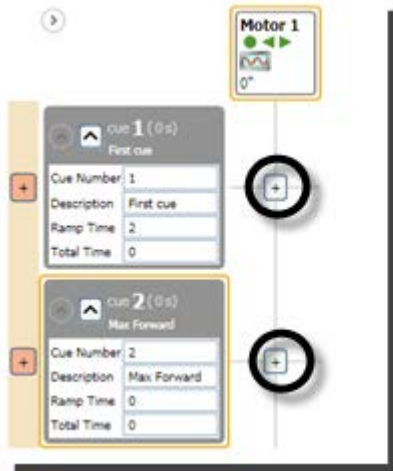


Figure 126

- After you press the “+” button, a movement will appear displaying the parameters for the motor in the cue (Figure 127).

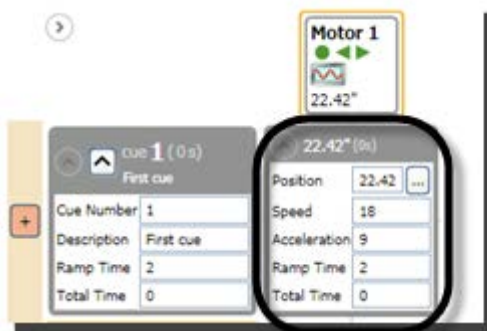


Figure 127

Motor Movement Parameters

Position (Figure 128) is the target position where you want the motor to go. This number is displayed in scaled units (inches, feet, degrees, etc.). By default, SpikeMark uses the current position of the motor as the target position. The button to the right of the **Position** text box will bring up the **Spike Selection** window. Spikes are discussed in the “Spikes” section on page 125.

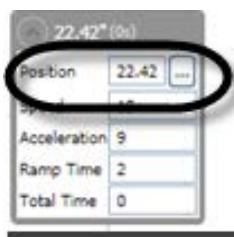


Figure 128

Speed (Figure 129) indicates the speed at which you want the motor to travel. This number is displayed in units per second (inches/sec, degrees/sec, etc.). You can synchronize multiple motors by programming them at the same speed, regardless of gearing and encoder differences. Valid speed values are greater than zero and no greater than the **Max Speed**, as defined in the **Speed** section of the **Motor Editing Pane**. SpikeMark initially uses the **Default Speed**, also defined in **Speed** section of the **Motor Editing Pane**.



Figure 129

Acceleration (Figure 130) indicates the rate at which the speed will increase as the motor moves. This number is displayed in units per second per second (inches/sec/sec, etc.). A higher value will cause the motor to accelerate more quickly giving the motion more snap at the beginning and end of the cue. A lower value will cause the motor to accelerate more slowly and smoothly. Valid acceleration values range from greater than zero to no greater than the programmed **Speed** value. SpikeMark initially uses the **Default Acceleration** defined in the **Speed** section of the **Motor Editing Pane**.

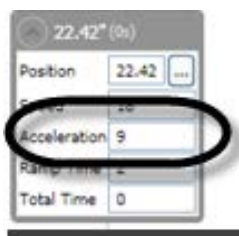


Figure 130

Ramp Time (Figure 131) indicates the number of seconds that the motor will take to accelerate up to programmed speed. If you would like to program the cue in time units, you can edit this field rather than using **Acceleration**. SpikeMark will automatically calculate a new **Acceleration** based on the **Ramp Time** you enter.

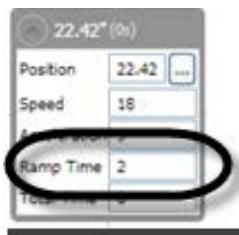


Figure 131

Total Time (Figure 132) indicates the total number of seconds that the movement will take to complete. If you know that the motor needs to move the scenery in 10 seconds, you can

enter a value of “10” here and SpikeMark will recalculate the **Speed** and **Acceleration** values to make that happen.

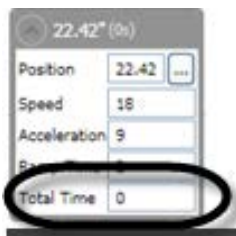


Figure 132

NOTE: SpikeMark computes the **Total Time** and **Ramp Time** in the first cue of the show by comparing the programmed position to the motor's current position. In all subsequent cues in the show, SpikeMark compares the programmed position to the *previous cue's* programmed position.

Changing Motor Position In a Cue

To change the programmed position of a motor in a cue you can do either of the following:

1. Type a new position value directly into the **Position** field (Figure 133).
*After you enter a new **Position**, SpikeMark will recalculate the **Total Time** of the movement based on the new target position. If you had previously specified a **Spike** for the movement it will be removed and replaced with the explicit value entered.*



Figure 133

Or, press the **Spike** button next to the **Position** box (Figure 134). Then select a **Spike** from the list and then click **OK** to accept (Figure 135).



Figure 134

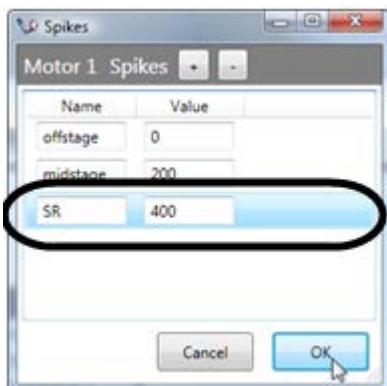


Figure 135

Changing How Fast a Motor Moves Within a Cue

There are two ways to adjust how fast a motor moves: Specify the *speed* of a motor directly or specify *how long* (the time) it should take for the motor to reach the target position and then let SpikeMark automatically compute the speed for you.

Specifying the Speed of a Motor Directly

1. Select the motor you wish to update in the **Cue Grid** and select the **Speed** value for the applicable motor movement (Figure 136).



Figure 136

2. Enter in a new **Speed** value and press the **Tab** key. (Figure 137).
*Note that the **Ramp Time** and **Total Time** are recomputed automatically by SpikeMark.*

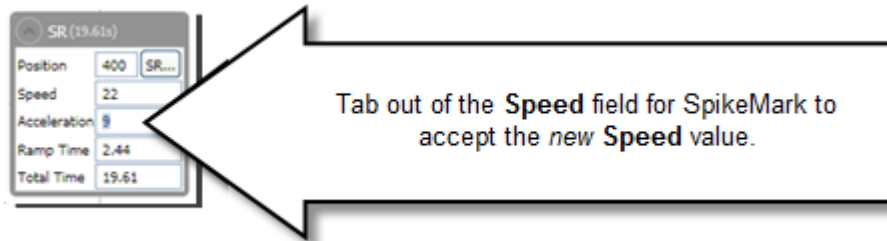


Figure 137

3. Enter in a new **Acceleration** value and press the **Tab** key. (Figure 138).
*Note that the **Ramp Time** and **Total Time** are recomputed automatically by SpikeMark.*

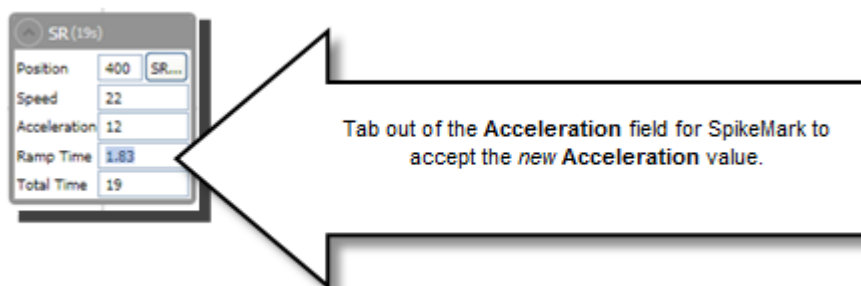


Figure 138

Specifying *How Long* the Motor Should Take to Reach the Target Position

1. Select the motor you wish to update in the **Cue Grid** and select the **Total Time** value for the applicable motor movement (Figure 139).



Figure 139

2. Enter in a new **Total Time** value and press the **Tab** key (Figure 140).
*Note that the **Speed** and **Acceleration** are recomputed automatically by SpikeMark.*

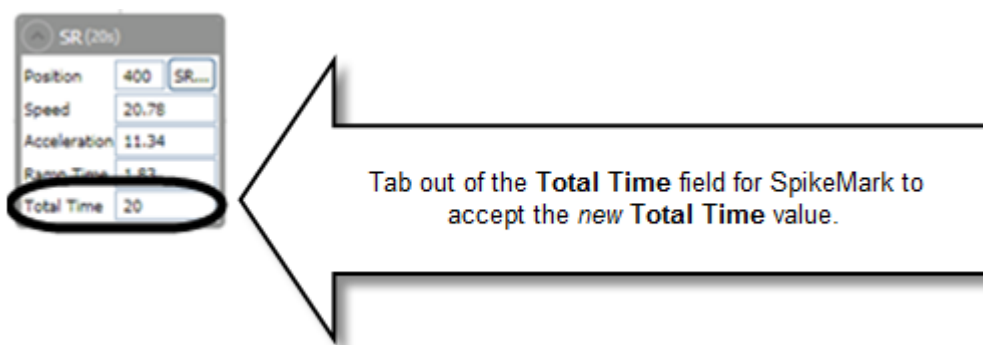


Figure 140

3. Select and change the **Ramp Time** value and press the **Tab** key (Figure 141).
*Note that the **Speed** and **Acceleration** are recomputed automatically by SpikeMark.*

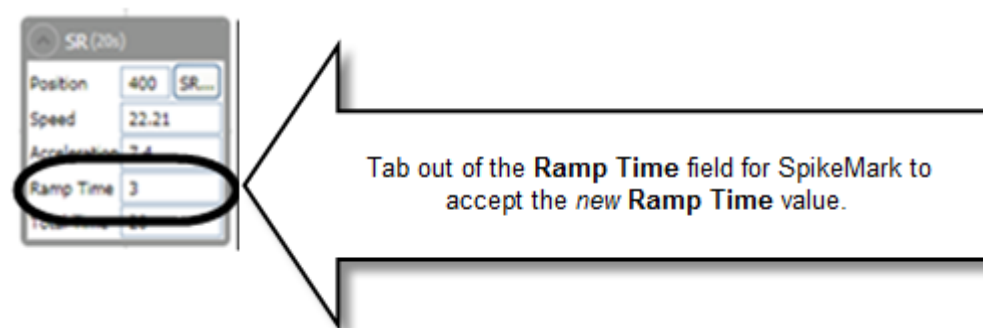


Figure 141

NOTE: If you enter a **Total Time** or **Ramp Time** combination that would require the motor to exceed its **Max Speed** [specified in the **Speed** section of the **Motor Editing Pane**], then the **Total Time** and **Ramp Time** will be recomputed to be *no higher than the Max Speed*.

Changing Cue Time

You can change the time for *all the movements in a cue* by specifying the **Total Time** and **Ramp Time** for the entire cue. This makes it easy to write cues that have all the motors ending at the same time.

1. Select the **Cue** you wish to edit in the **Cue Grid** (Figure 142).



Figure 142

2. Type in the new **Total Time** value you wish to use and press the **TAB** key (Figure 143).

*Note that all the **Total Time** values for each movement have been updated.*



Figure 143

3. Repeat the steps for **Ramp Time**, if desired.

Adding an FX Movement to a Cue

Adding an FX movement to a cue is similar to adding a motor movement.

1. Click the “+” button in the **Cue Grid** correct associated with the Stagehand FX and Cue you wish to edit (Figure 144).

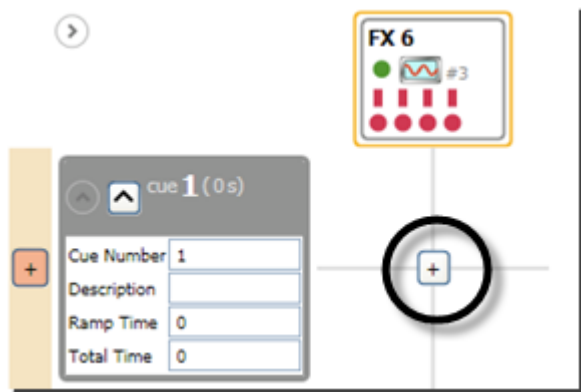


Figure 144

2. An FX movement will appear in the **Cue Grid** (Figure 145).

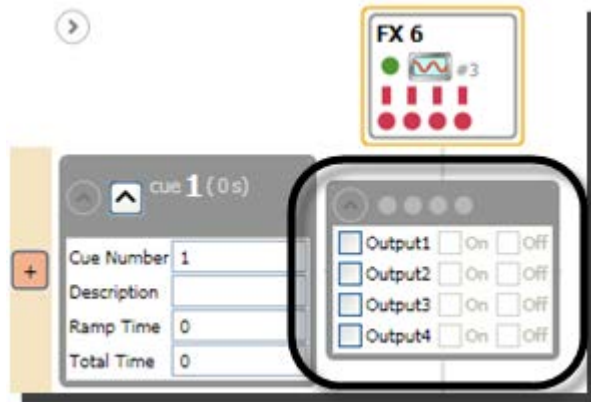


Figure 145

NOTE: An FX movement can turn any output switch on or off in the cue. If a switch is turned on in one cue, it will remain on **until another cue turns it off**. Therefore, if you don't want to change the state of a switch in a cue, leave the switch unchecked completely.

Figure 146 is an example series of cues that turns Output 1 on, then off, then turns Outputs 2, 3 and 4 on, while leaving Output 1 off.

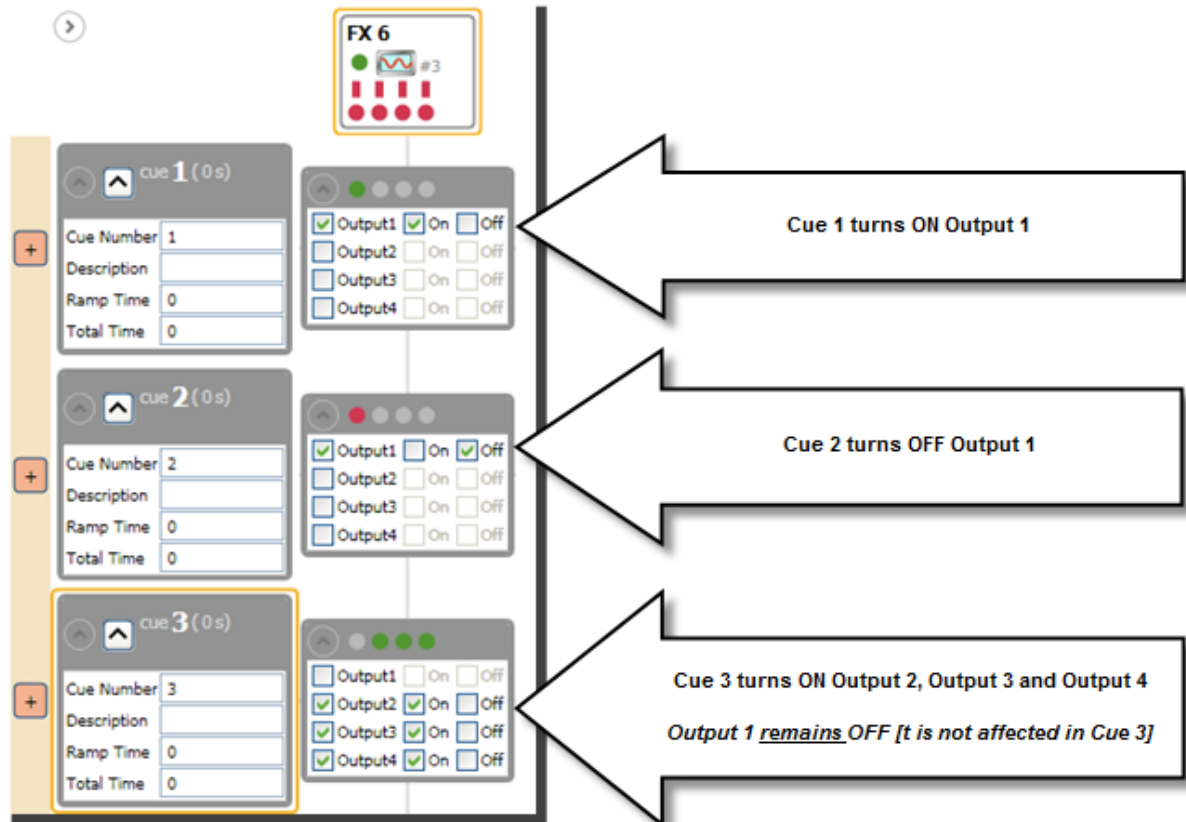


Figure 146

Removing a Motor's Movement from a Cue

Sometimes you will write a cue with a number of motors moving and then discover one of the motor movements should not be in the cue.

1. Right-click on the motor's movement you wish to delete in the **Cue Grid** and select **Delete Movement** (Figure 147).



Figure 147

2. Click **Yes** in the **Remove Movement** confirmation dialog box (Figure 148).

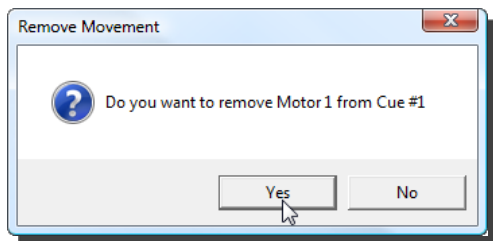


Figure 148

Deleting a Cue

Theatre is a dynamic process. Throughout rehearsals scene shifts will be added, and scrapped as the performance is shaped. To keep up with the rapid changes, SpikeMark makes it just as easy to remove cues as it is to add them. There are two simple ways to delete a cue from a show:

1. Right-click on the cue you wish to delete in the **Cue Grid** and select **Remove Cue** (Figure 149).

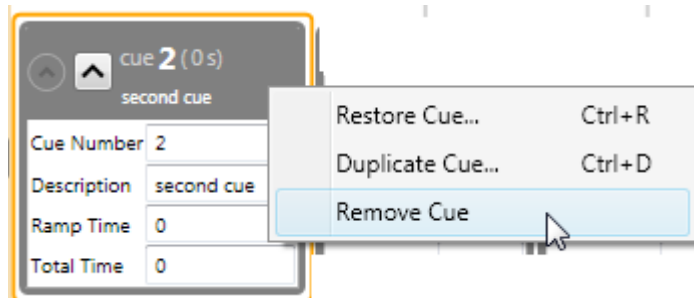


Figure 149

2. Click **Yes** in the **Remove Cue** confirmation dialog box (Figure 150).

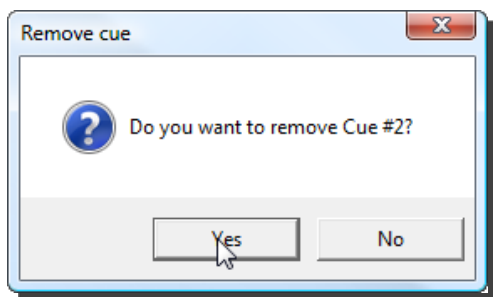
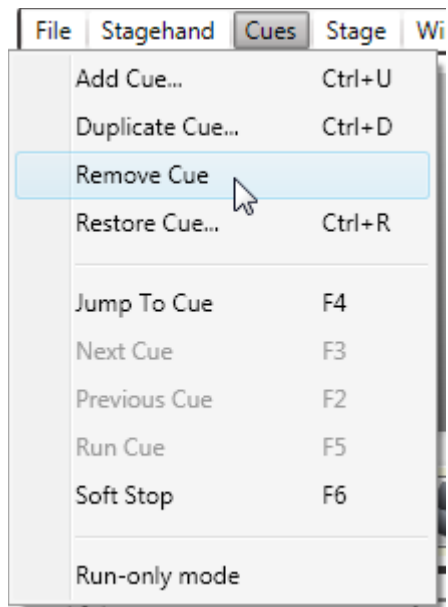


Figure 150

Or, select the cue you wish to delete in the **Cue Grid**. In the **Cues** menu, click **Remove Cue** (Figure 151) and then click **Yes** to confirm the deletion.

**Figure 151**

Cue Links

As you are writing cues in a new show, you'll find scene changes in which simple cues fall short of your desired effect. You might want to fire a sequence of movements, all within in one cue, but each with a *different* start time. For example, you might want a two-second delay between the start of each movement. Or, maybe the house wagon needs to wait to move upstage until the balcony unit is clear and in the wings. Whatever the need, links allow you to create sequences of cues that start automatically in a staggered fashion.

Links are great for cue sequences that can be reliably run in the exact same way every performance. If you find that a link is too rigid (for example, you need to run two cues, timed such that the second cue runs right after an actor says a specific line), a cue link might be too rigid, because the timing of the actor's line is unpredictable from night to night. In these instances, you should keep the cues independent or, use **Pile On** cues. Pile On cues are loaded *manually*, and at your own discretion, before the previous cue completes.

When discussing **Cue Links**, we need a vocabulary to refer to the two cues that are linked together:

Parent Cues are the cues being watched by the link. The **Parent Cue** is run by either pressing the GO button, or it may be a Child Cue (see below) of another link in a longer sequence of linked cues. Whenever the Parent Cue is running, the link will check its criteria and automatically start the **Child Cue** when the criteria are met.

Child Cues are the cues that are started by the link. When a link's criteria are satisfied, the link will automatically run the **Child Cue**. Graphically, Child Cues have the link displayed next to them in the **Cue Grid**.

All links are displayed in the left margin of the **Cue Grid** (Figure 152).

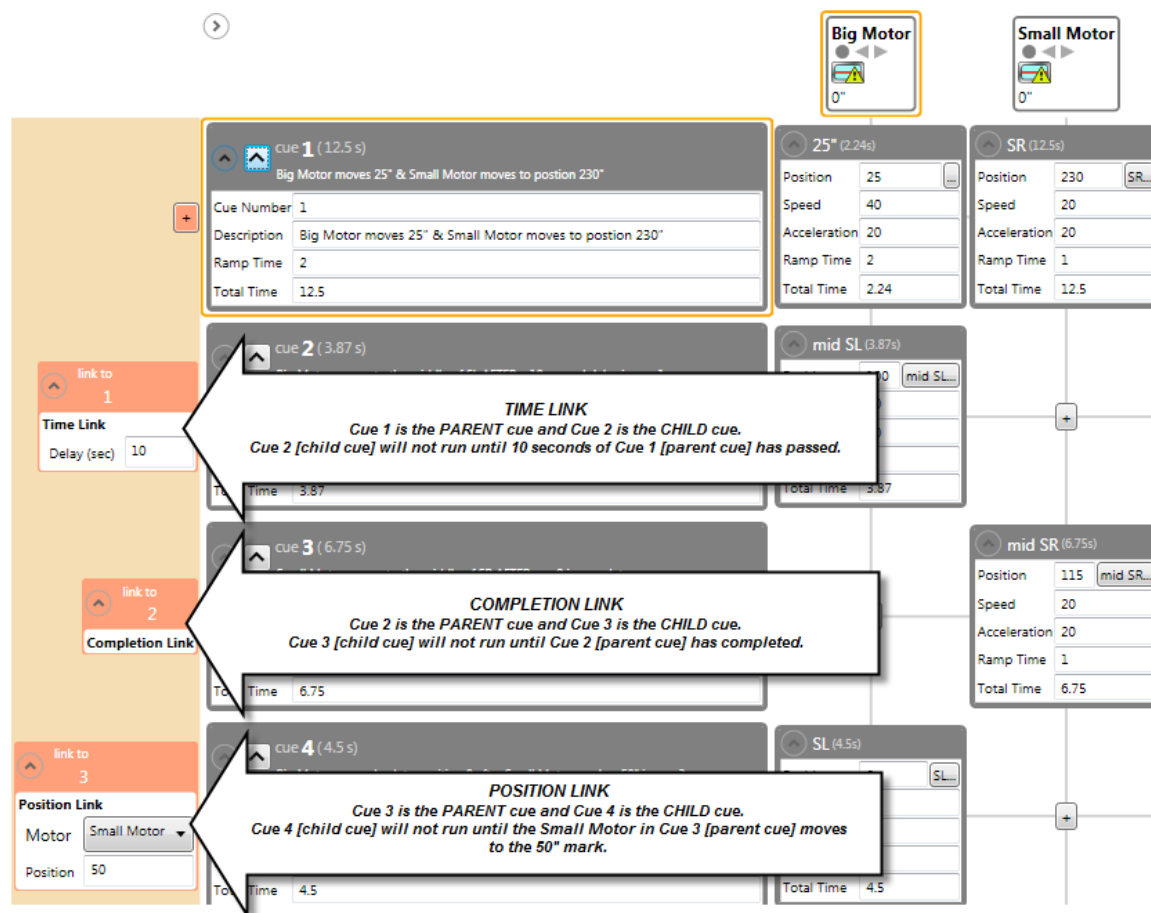


Figure 152

NOTE: You can create an infinite loop by creating two links between two different cues. You would set up your links so that Cue 1 is the **Parent Cue** of Cue 2 and Cue 2 is the **Parent Cue** of Cue 1. This might be useful if you are testing machines OR if you need a motor to cycle back and forth indefinitely. To stop such a loop of cues, you would click the **Stop** button in the **Cue Controller**, or use the **Emergency Stop** on the Showstopper or use a “Soft Stop Motor” Input Action with a Stagehand FX.

Linking Cues by Completion

You can link two cues together by completion, so that when the **Parent Cue** completes successfully, the **Child Cue** will begin.

1. Click the **Add Link** button next to the **Child Cue** (Figure 153).

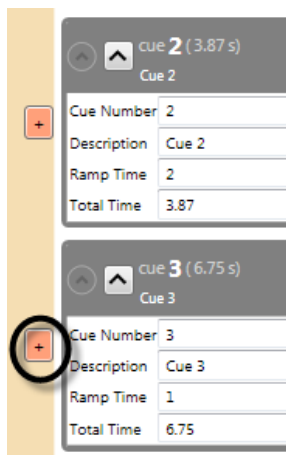


Figure 153

2. Select the **Parent Cue** from the drop down list and select **Completion Link**, then press **OK** (Figure 154).

*In the example pictured below, the selected **Parent Cue** is “Cue #2.”*

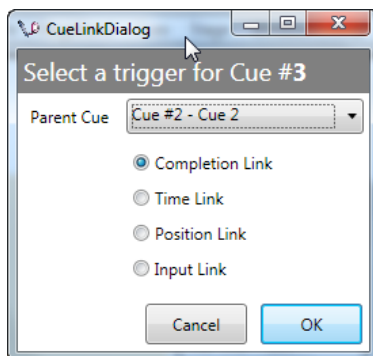


Figure 154

3. The new link will appear in the left margin of the **Cue Grid** (Figure 155).
In the example below, Cue 3 (Child Cue) will not run until Cue 2 (Parent Cue) has completed.

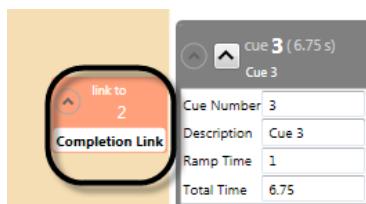


Figure 155

Linking Cues by Time

You can link two cues together with a time delay so that the **Child Cue** runs a programmed number seconds after the **Parent Cue** starts.

1. Click the **Add Link** button next to the **Child Cue** (Figure 156).

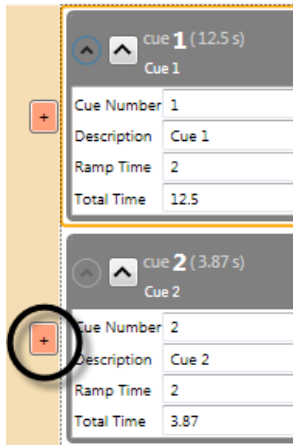


Figure 156

2. Select the **Parent Cue** from the drop down list and select **Time Link**, then press **OK** (Figure 157).

*In the example pictured below, the selected **Parent Cue** is “Cue #1.”*

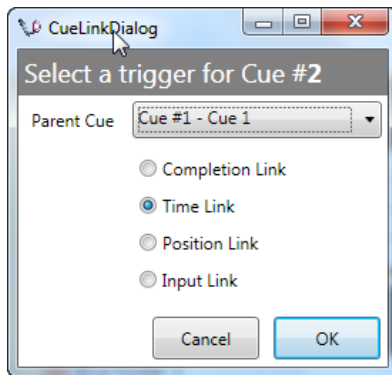


Figure 157

3. The new link is displayed in the left margin. You can edit the **Delay** value (Figure 158).
In the example below, Cue 2 (Child Cue) will not start until Cue 1 (Parent Cue) has run for 10 seconds.

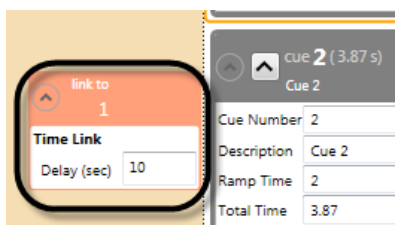


Figure 158

Linking Cues by Motor Position

You can link a **Child Cue** so that it runs when one of the motors in the **Parent Cue** reaches a defined *position*. This is handy when you have to ensure that one motor has cleared the way for another motor.

1. Click the **Add Link** button next to the **Child Cue** (Figure 159).

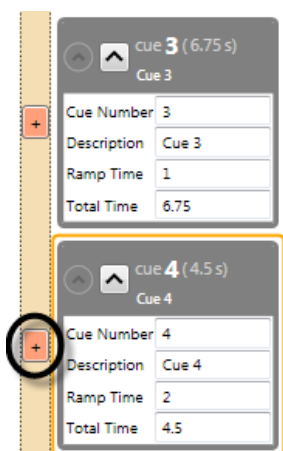


Figure 159

2. Select the **Parent Cue** from the drop-down list and select **Position Link**, then press **OK** (Figure 160).

*In the example pictured below, the selected **Parent Cue** is “Cue #3.”*

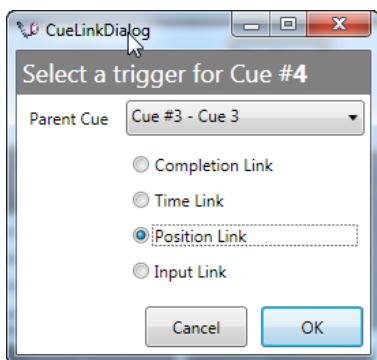


Figure 160

3. The new link will appear in the left margin next to the **Child Cue**. Select the **Motor** from the **Parent Cue** and enter a position that will trigger the link (Figure 161).

*In the example below, Cue 4 (Child Cue) will not start until the **Small Motor** in Cue 3 (Parent Cue) reaches the 50" mark.*

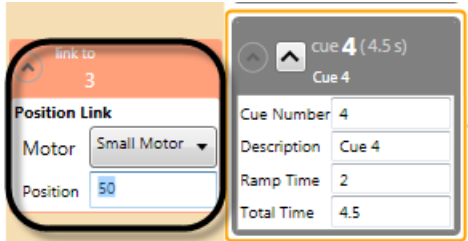


Figure 161

Linking Cues by FX Input

You can link a **Child Cue** to a **Parent Cue** so it runs as soon as an input switch is detected from a Stagehand FX. This can be handy if you have two cues that need to run in close proximity, but require a crew member to give a “Clear” signal for the second cue. In such a case, you could wire a pushbutton to the input of a Stagehand FX. When the **Parent Cue** runs, the crew member would push the button to start the **Child Cue** when appropriate.

1. Click the **Add Link** button next to the **Child Cue** (Figure 162).

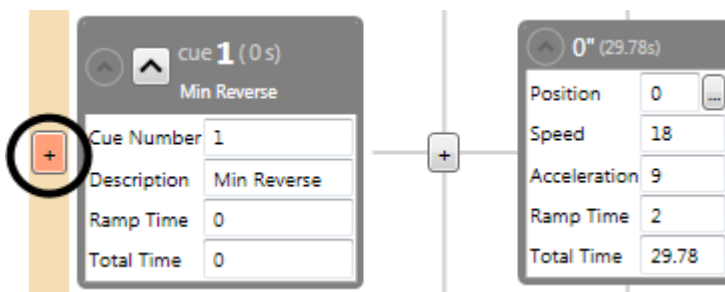


Figure 162

2. Select the **Parent Cue** from the drop-down list and select **Input Link**, then press **OK** (Figure 163).

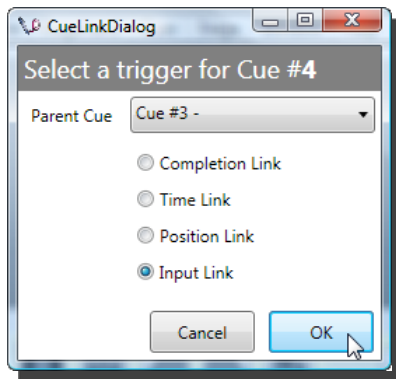


Figure 163

3. The new link will appear in the left margin of the **Cue Grid**. Select the Stagehand FX and the input that should trigger the **Child Cue** (Figure 164).

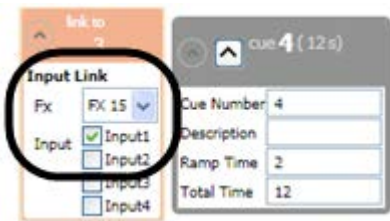


Figure 164

Deleting a Cue Link

1. Right-click on the **Link** in the **Cue Grid** and select **Delete Link** (Figure 165).

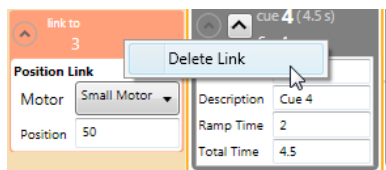


Figure 165

2. Click **Yes** in the **Remove Link** confirmation dialog box (Figure 166).

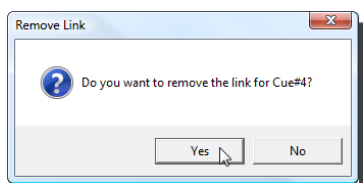


Figure 166

Multi-Speed/Multi-Position Links

There are some cues that require a motor to change *speed* in the middle of the movement. Perhaps a turntable needs to move at a walking pace for 30 seconds and then speed up to running pace and then back down.

Other cues require the motor to oscillate smoothly between two *positions* without coming to a stop and setting the brake, such as a wave which rises and falls smoothly throughout a scene.

SpikeMark can handle either of these scenarios with ease. The steps are illustrated in the sections below: “Changing the Speed of a Motor—Manually,” “Changing the Speed of a Motor—Automatically” and ““Ping-Pong” a Motor—Changing a Motor’s Position.”

It should be noted that it is feasible to change *both* the speed and the position of a motor, so that your wave might move between two positions, rising up and down and change speed, moving slowly and then faster.

Changing the Speed of a Motor—Manually (Without a Link)

1. Write two cues for the same motor, with the same target **Position**.
In Figure 167, the target **Position** in each cue is “0.”

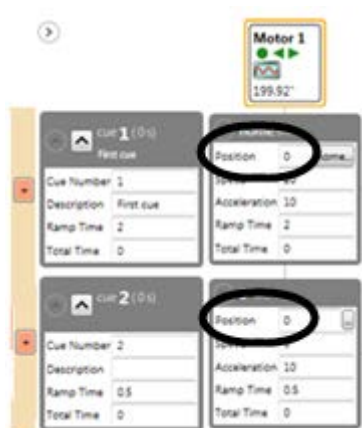


Figure 167

2. In each Cue, enter the two different **Speed** values.
In the example in Figure 168, the **Speed** in Cue 1 is “20” and the **Speed** in Cue 2 is “5.” The **Acceleration** is controlled by the first cue, and cannot be altered mid-move.



Figure 168

3. While Cue 1 is running, load Cue 2 and run it whenever you want the motor to decelerate to the new speed.

Changing the Speed of a Motor—Automatically

1. Write two cues for the same motor, with the same target **Position**.
In the example in Figure 169, the target **Position** in each cue is “0.”



Figure 169

2. In each cue, enter the two different **Speed** values.
In the example in Figure 170, the **Speed** in Cue 1 is “20” and the **Speed** in Cue 2 is “5.” The **Acceleration** is controlled by the first cue, and cannot be altered mid-move.

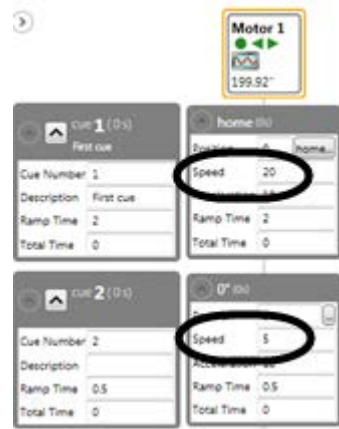


Figure 170

3. Link the two cues together using a **Time Link** or **Position Link**.
*In the example in Figure 171, Cue 1 and Cue2 are linked with a **Time Link**.*



Figure 171

NOTE: When linking two cues to automatically change a motor's speed, do NOT use a **Completion Link**, as this will bring the motor to a complete stop. Use a **Time Link** or **Position Link** in order to smoothly change speeds.

“Ping-Pong” a Motor—Changing a Motor's Position

Creating a “Ping Pong” Motor effect creates an “infinite” loop, where the motor travels between two positions *without coming to a hard stop*. Instead the motor gracefully decelerates to zero-speed, changes direction, and then immediately accelerates in the opposite direction.

Since the “Ping-Pong” affect creates an “infinite” loop, you will have to interrupt the sequence by using any of the following: the **Stop** on the **Cue Controller**; the **Stop All Cues** or **Emergency Stop** buttons on the Showstopper; or a “Soft Stop Motor” Input Action with a Stagehand FX.

1. Write two cues with different **Position** targets for the same motor.
*In the example in Figure 172, the **Position** targets are “0” and “100.”*

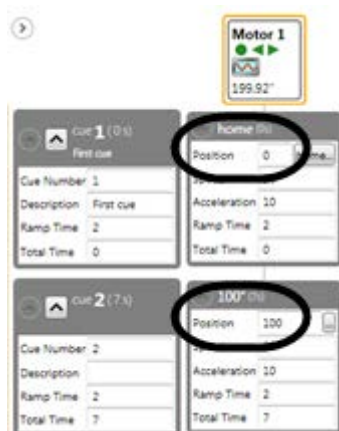


Figure 172

2. Link the two cues together with a **Position Link** or **Time Link**.
*In the example in Figure 173, Cue 1 links to Cue 2 and Cue 2 links to Cue 1, with **Position Links***



Figure 173

NOTE: When linking two cues to create a "ping-pong" effect, do NOT use a **Completion Link**, as this will bring the motor to a complete stop. Use a **Time Link** or **Position Link** in order to slow the motor to an imperceptible stop, creating a more fluid look for the position change.

The Stage Model Viewer

SpikeMark's **Stage Model Viewer** (Figure 174) provides the perfect way to visualize your show. Rather than seeing the bland rows and columns of numbers in the **Cue Grid**, the **Stage Model Viewer** shows you a three-dimensional schematic of your stage. Every motor in the show can be sized and positioned on the virtual stage so it closely resembles the actual scenery in your theatre.

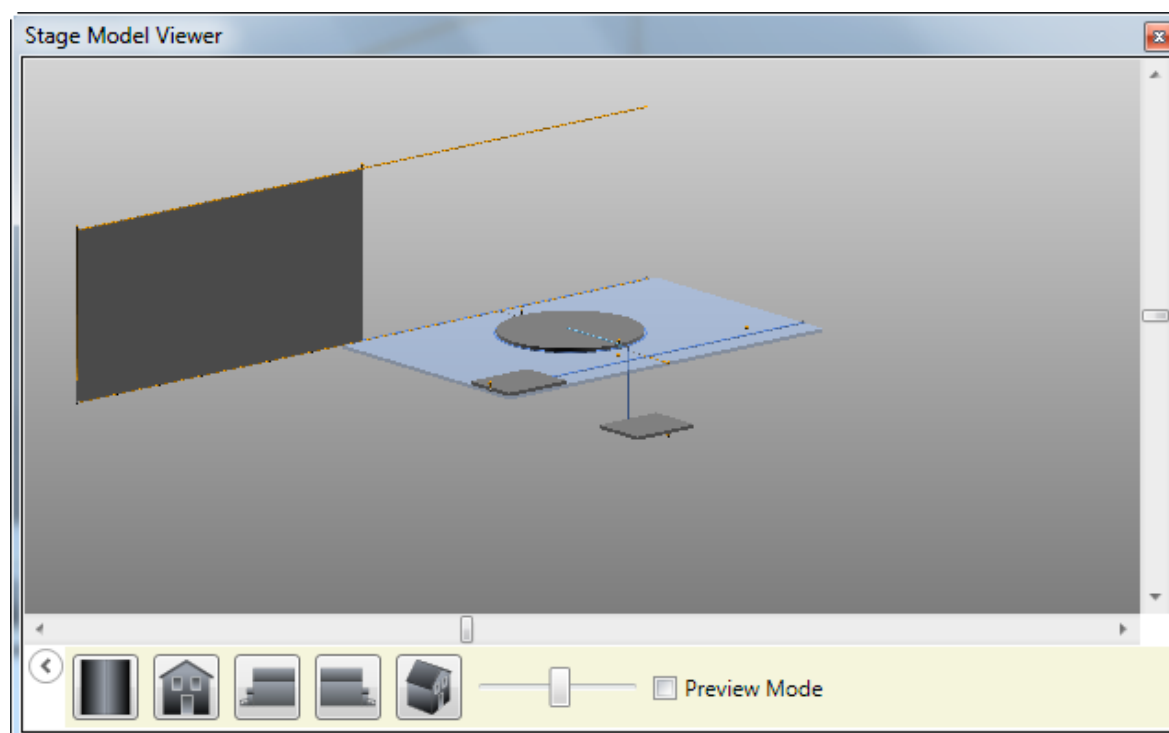


Figure 174

When you load a cue, any motor that is programmed to move will change color and display its destination in the **Stage Model Viewer**, allowing you to see—at a glance—what will move and where it will move to. As the motors move in real life, they move in the **Stage Model Viewer** as well, giving you instant feedback about what is happening on the stage. This feedback is useful when you are sitting back stage trying to keep track of every piece of moving scenery and every person on the stage, which can make it much easier to quickly decide if someone is in harm's way before the Stage Manager shouts, "GO!"

Another great way to use the **Stage Model Viewer** is as a pre-tech cue-writing tool. Using **Spikes** (also see the "Spikes" section on page 125) and the **Stage Model Viewer** you can rough-in your cues and clearly see what the stage will look like before setting foot in the theatre. This can give you and the artistic staff time to play with different looks before tech and speed up the rehearsal process. In this section, we'll examine how to use the **Stage Model Viewer** as a pre-

tech cue-writing tool. For more information on how to use the **Stage Model Viewer** when you are running shows, see the “Running Shows” section on page 138.

NOTE: To open the **Stage Model Viewer** in a separate window, click **Stage Model Viewer** in the **Window** menu.

Navigating the Stage Model Viewer

You can use your right mouse button to pan around the **Stage Model Viewer** space by clicking and dragging inside the 3D model. You can also use the on-screen tools—listed below—to zoom, pan, and twirl the 3D model.

The **Zoom Slider** (Figure 175) allows you to magnify the **Stage Model Viewer**. You can also use the scroll-wheel on your mouse to zoom in or out.

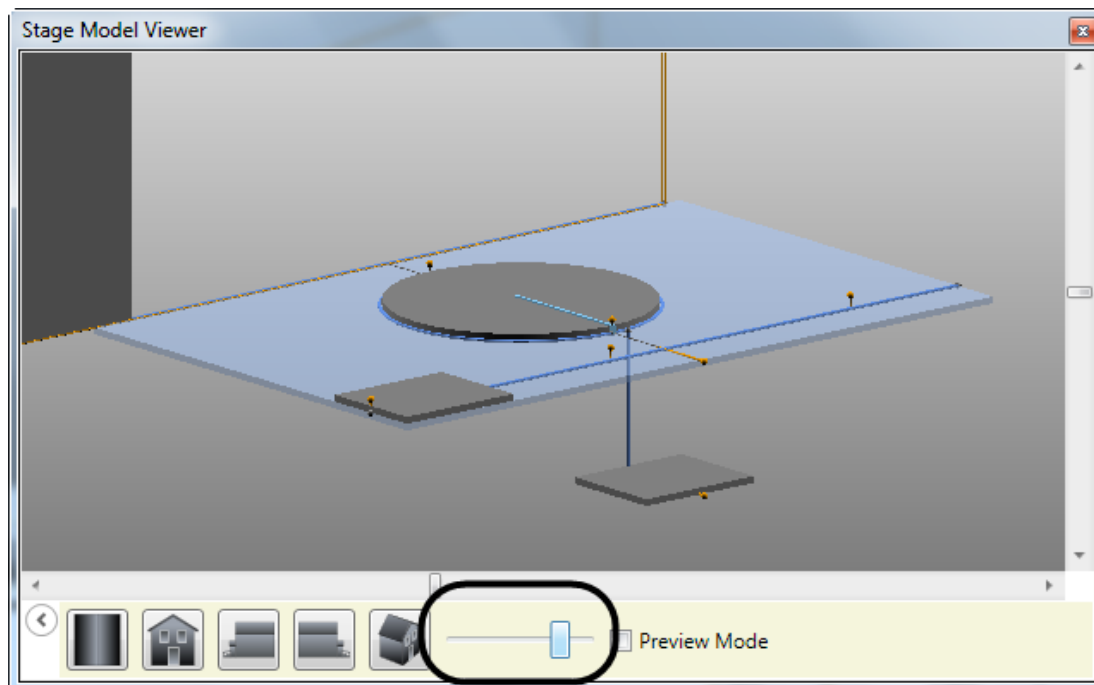


Figure 175

Use the **Scrollbars** (Figure 176) on the right and bottom of the **Stage Model Viewer** to orbit around the stage.

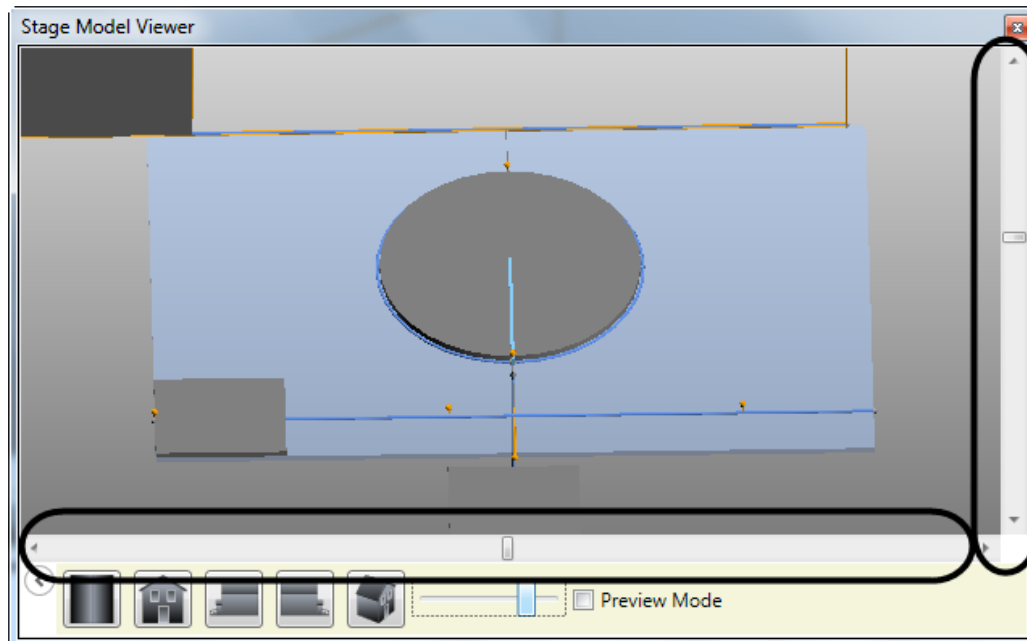


Figure 176

The **Preset View Buttons** allow you to quickly jump to the 5 orthographic projections (Figure 177 through Figure 181).

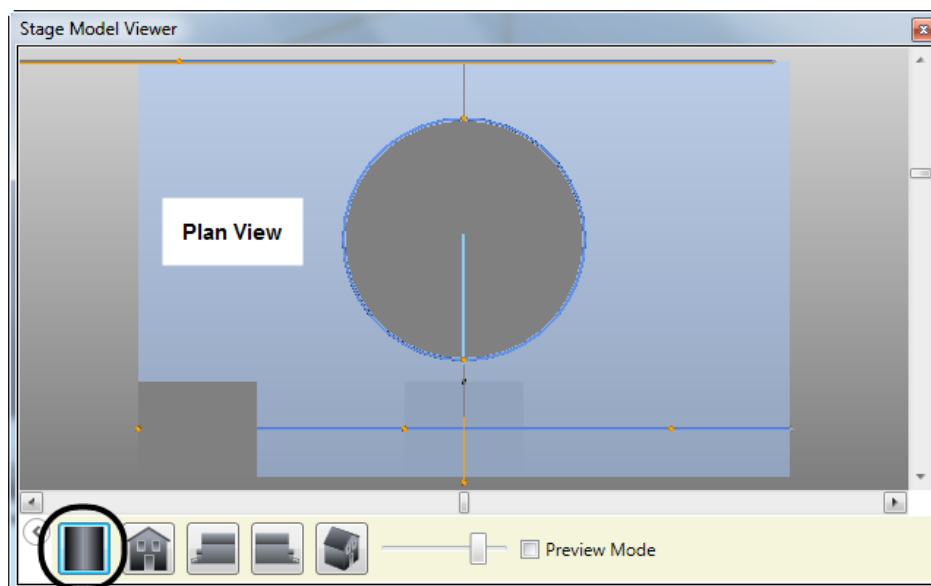


Figure 177

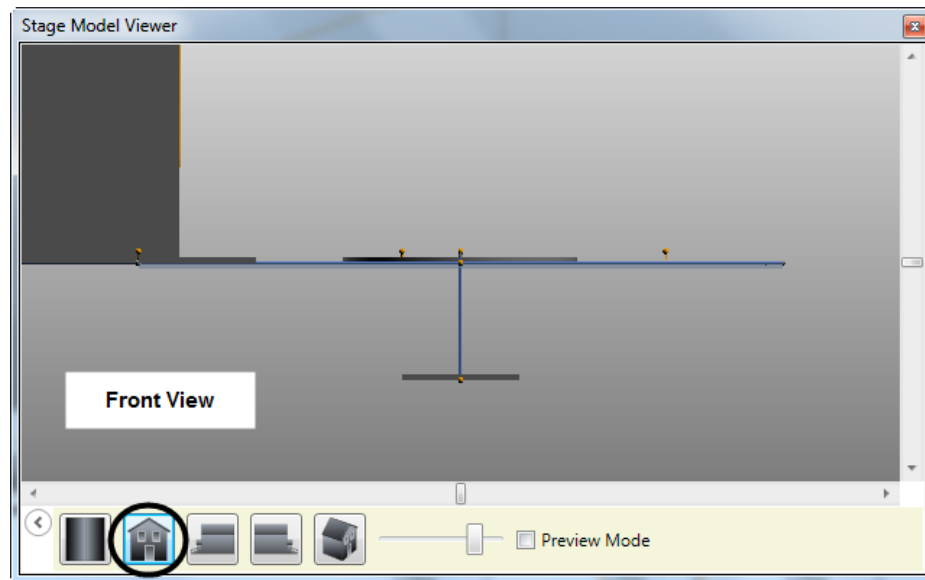


Figure 178

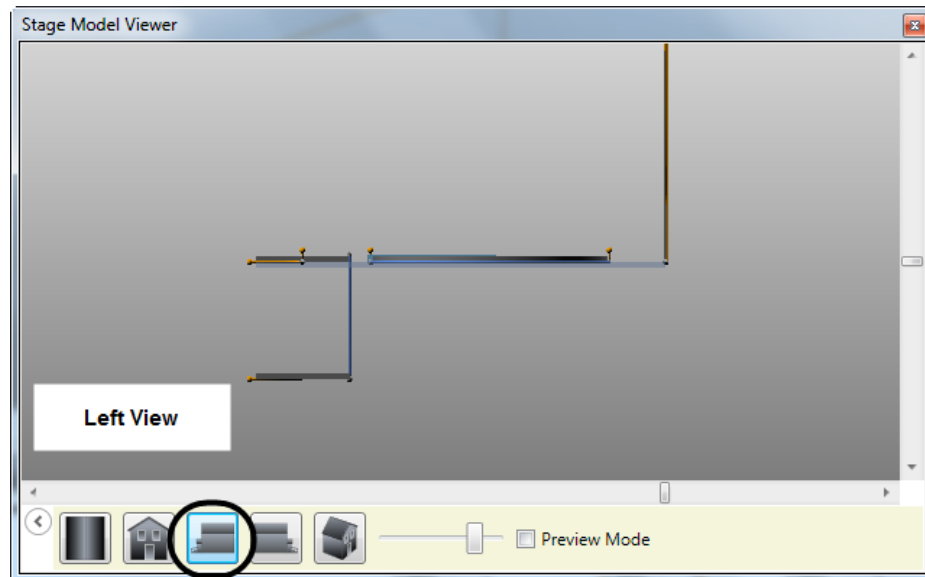


Figure 179

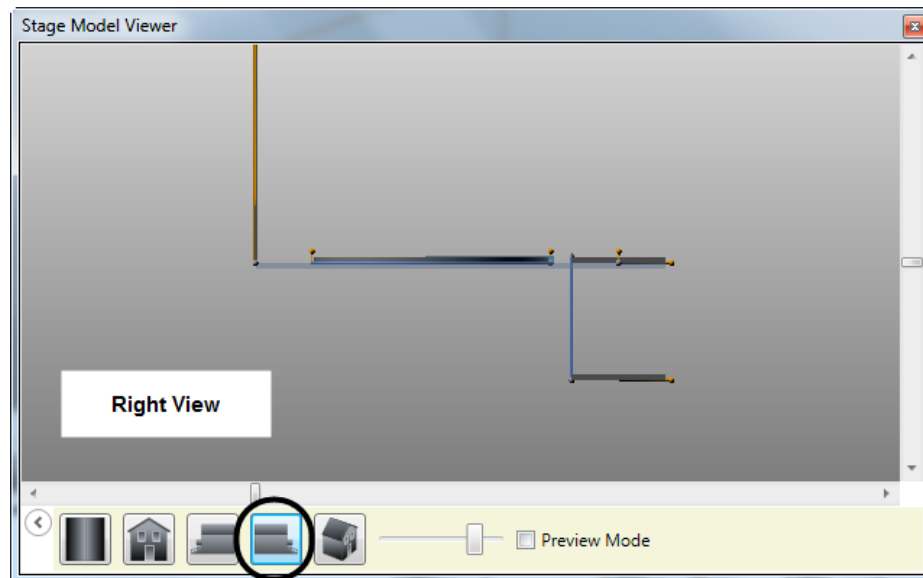


Figure 180

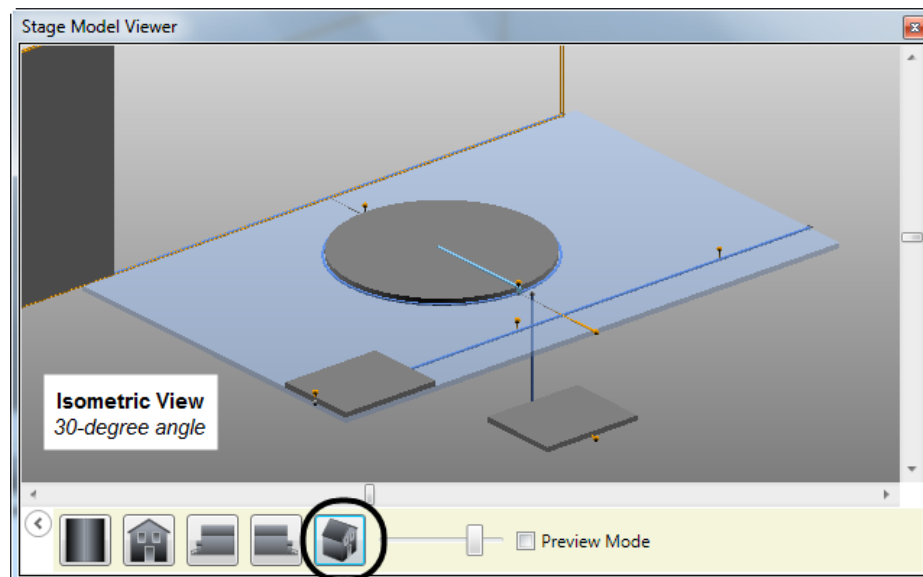


Figure 181

Setting the Stage Dimensions

In the **Stage Model Viewer**, the stage is shown as a semi-transparent box. To make the stage in the **Stage Model Viewer** more closely resemble the size of your stage, you can change the default stage dimensions.

1. From the **Stage** menu, click **Stage Dimensions** (Figure 182).

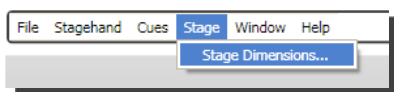


Figure 182

2. Edit the dimensions of the stage to match your physical stage and click **OK** (Figure 183). *All Stage Dimensions are in feet.*

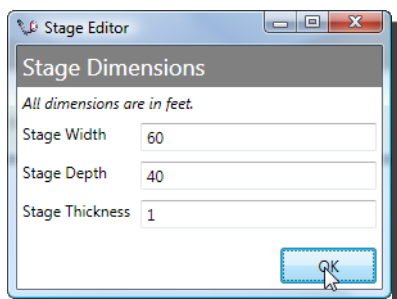


Figure 183

Making a Winch Model

After you've changed the size of the stage to match your theatre space, the next step is to make the motors look roughly like the scenery on the stage. Each motor in the show can be viewed as either a winch or a turntable. Winches always look like a box traveling along a track. The length of the track is determined by the **Max Forward** and **Min Reverse** parameters (set in the **Position** section of the **Motor Editing Pane**). The size and placement of the scenery on the track can be adjusted in the **Schematic** section of the **Motor Editing Pane**.

1. Select the motor you wish to adjust in the **Cue Grid**.
2. In the **Motor Editing Pane**, scroll to the **Schematic** section and make sure that **Winch** is selected at the bottom (Figure 184).
Adding a Turntable is described in the "Making a Turntable Model" section on page 121.

Schematic

All dimensions are in feet.

Distance from Center

Distance from Plaster

Elevation from stage

Width

Length

Height

Rotation X

Rotation Y

Rotation Z

☒ winch
☐ turntable

☒ Turntable Clockwise Positive

Figure 184

3. Adjust the **Width**, **Length**, and **Height** properties (Figure 185) by using either the sliders beneath each text field or typing in a dimension directly. As you change the dimensions, the winch model in the **Stage Model Viewer** will update.
Note that all dimensions are given in feet.

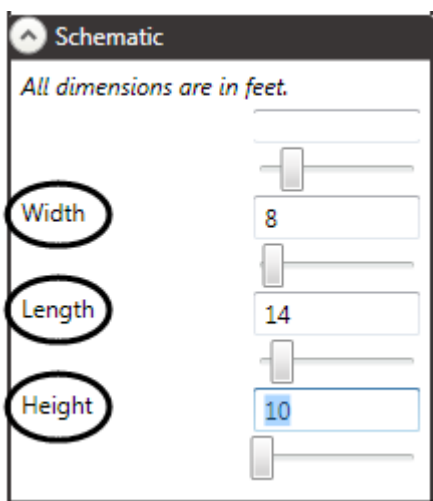


Figure 185

4. Now, move the winch into position so that it matches your stage. Use the **Distance from Center**, **Distance from Plaster** and **Elevation from stage** field sliders (Figure 186) to get the position close. You can then fine tune the position by typing in accurate dimensions from your ground plan.

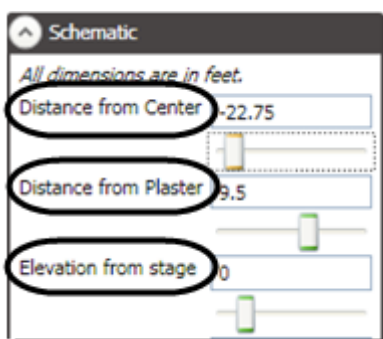


Figure 186

NOTE: By default, a winch is displayed with its minimum reverse position **Stage Right** [the **Min Rev Position** is set in the **Position** section of the **Motor Editing Pane**] and its forward direction heading **Stage Left**. You can use the **Rotation Y** parameter [in the **Schematic** section of the **Motor Editing Pane**] to spin the winch around and reposition it if you need the **Min Rev Position** to be **Stage Left** and its forward directing heading **Stage Right**. (See Figure 187, Figure 188 and Figure 189).

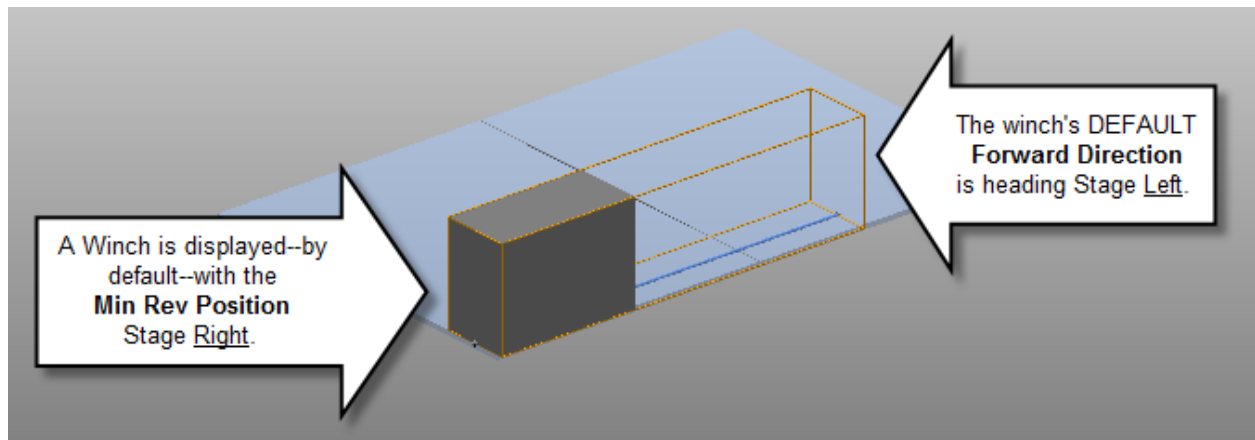


Figure 187

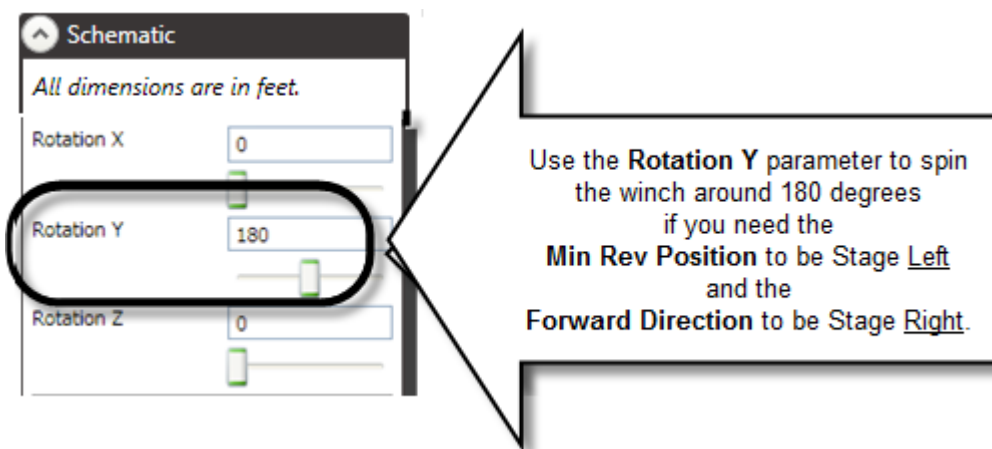


Figure 188

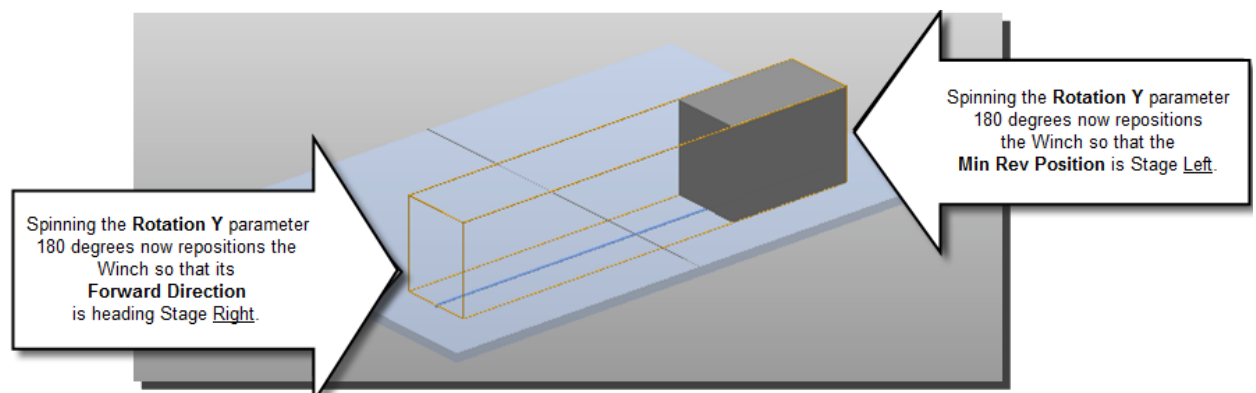


Figure 189

Making a Lift Model

The winch-style model can be used to represent various machines other than ordinary deck winches. By adjusting the size and rotation of the winch we can easily make it look like a Lift.

1. Select the motor you wish to adjust in the **Cue Grid**.
2. In the **Motor Editing Pane**, scroll until you find the **Schematic** section and make sure that **Winch** is selected at the bottom (Figure 190).
Adding a Turntable is described in the “Making a Turntable Model” section on page 121.

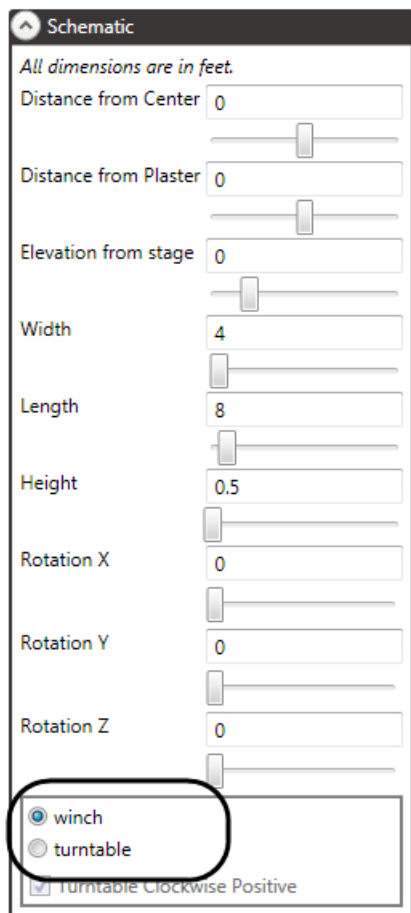


Figure 190

3. Using the **Rotation X** and **Rotation Z** sliders, rotate the winch model until it is standing upright (Figure 191).

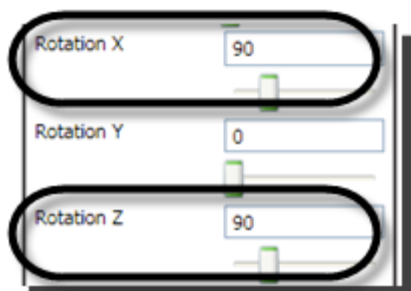


Figure 191

4. Now use the **Width**, **Length**, and **Height** sliders to make the winch look like a lift platform (Figure 192 and Figure 193).

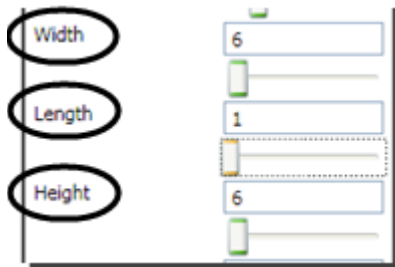


Figure 192

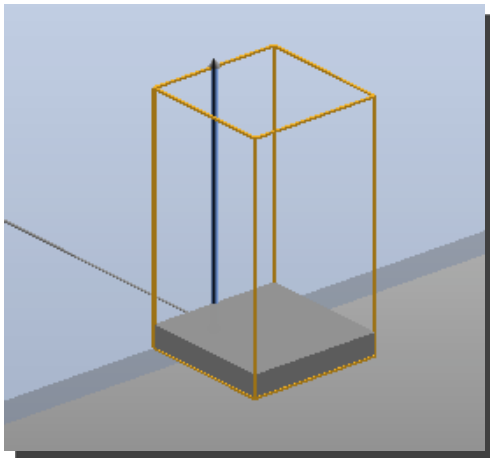


Figure 193

5. Using the **Elevation from stage** field, position the Lift so that it travels into the trap room (Figure 194 and Figure 195).
Use the **Distance from Center** and **Distance from Plaster** fields to place your Lift in the appropriate part of your stage.

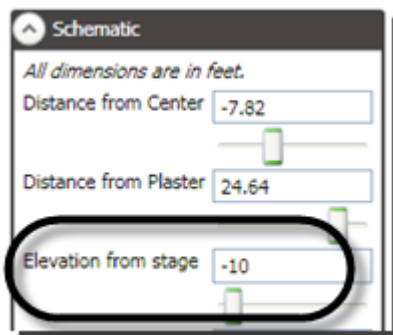


Figure 194

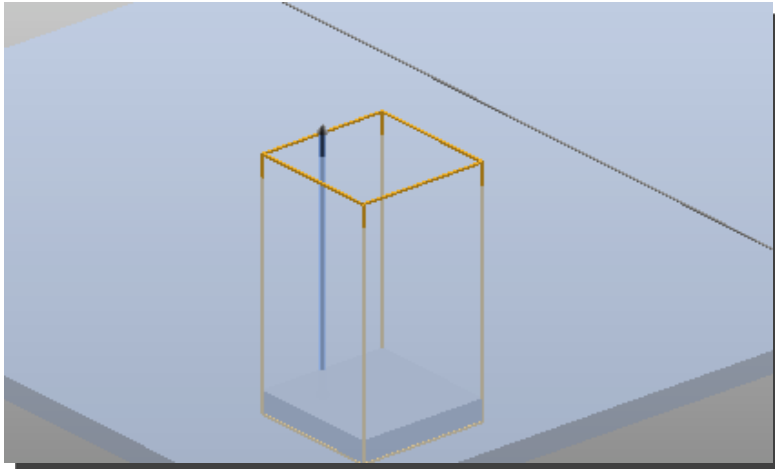


Figure 195

NOTE: The stage is semi-transparent so that you can see scenery moving, even when it is beneath the stage.

Making a Curtain Model

The winch-style model can be used to represent various machines other than ordinary deck winches. By adjusting the size and rotation of the winch, we can easily make it look like either a traveling curtain or a flying curtain.

1. Select the motor you wish to adjust in the **Cue Grid**.
2. Scroll to the **Schematic** section of the **Motor Editing Pane** and make sure that winch is selected at the bottom of the list (Figure 196).
Adding a Turntable is described in the Making a Turntable Model section on page 121.

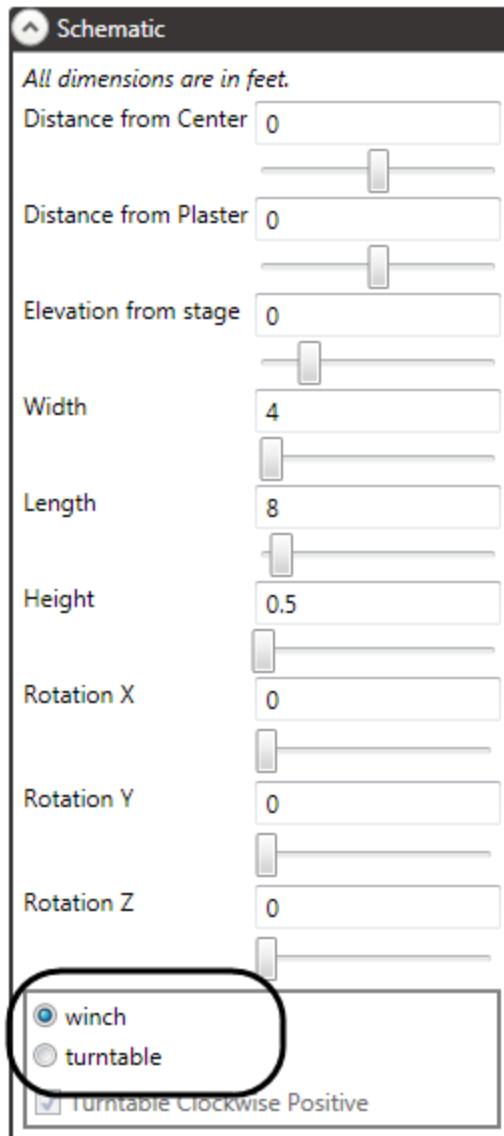


Figure 196

- Adjust the **Width**, **Length**, and **Height** properties to make a tall, thin curtain (Figure 197 and Figure 198).



Figure 197

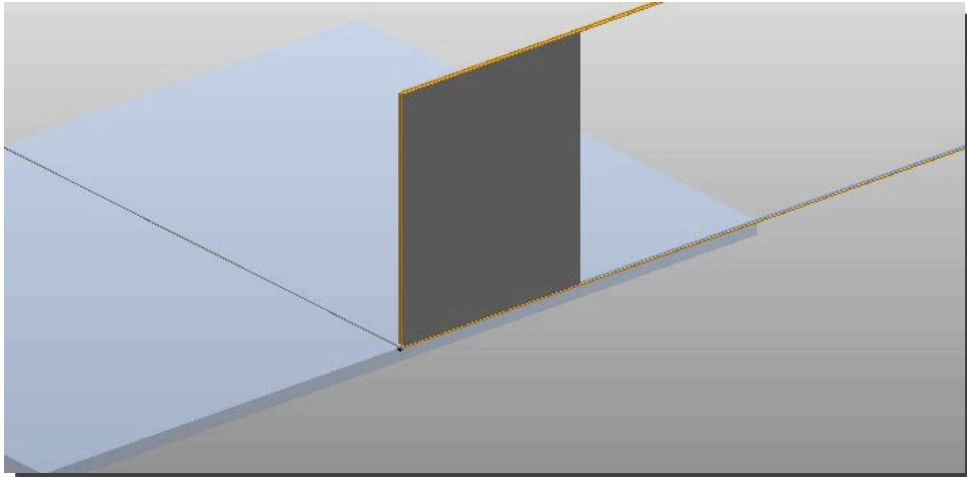


Figure 198

4. Using the **Distance from Center**, **Distance from Plaster** and **Elevation from Stage** fields, position the curtain on the stage (Figure 199 and Figure 200).

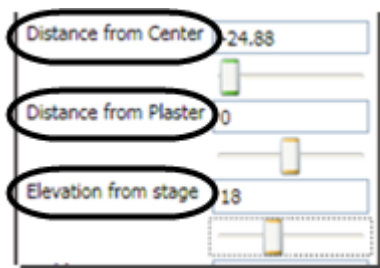


Figure 199

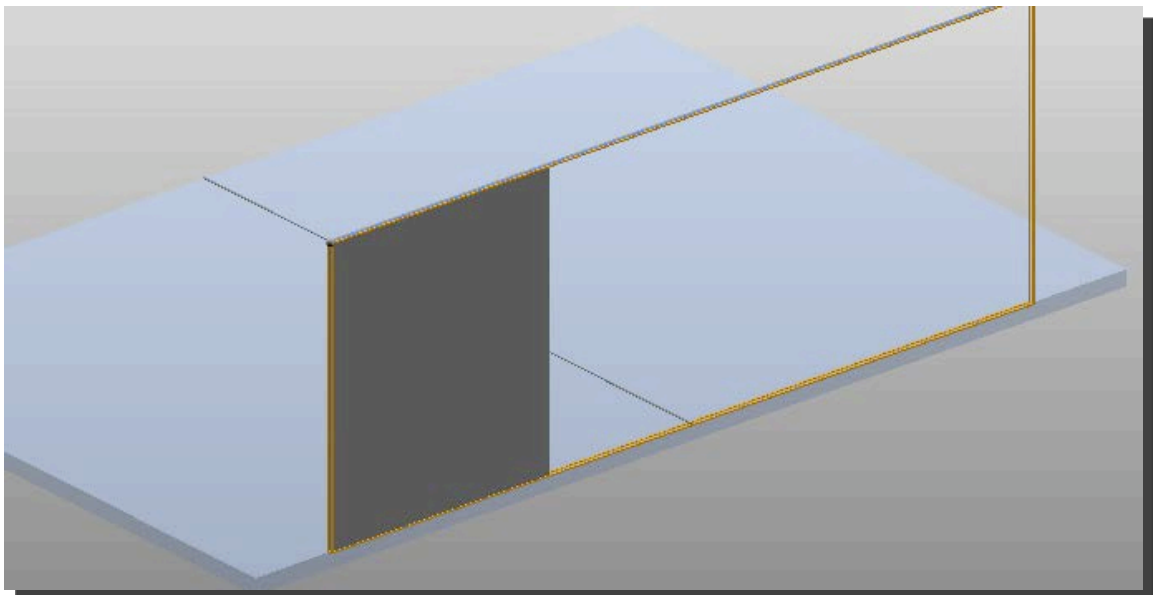


Figure 200

5. By adjusting the **Rotation Z**, **Distance from Center**, **Distance from Plaster**, **Length** and **Height** fields, you can change the horizontal traveler curtain into a vertical flying curtain (Figure 201 and Figure 202).

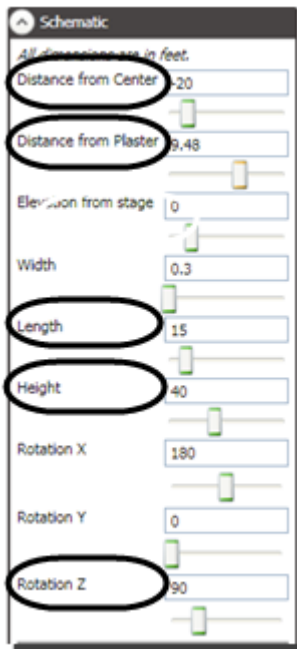


Figure 201

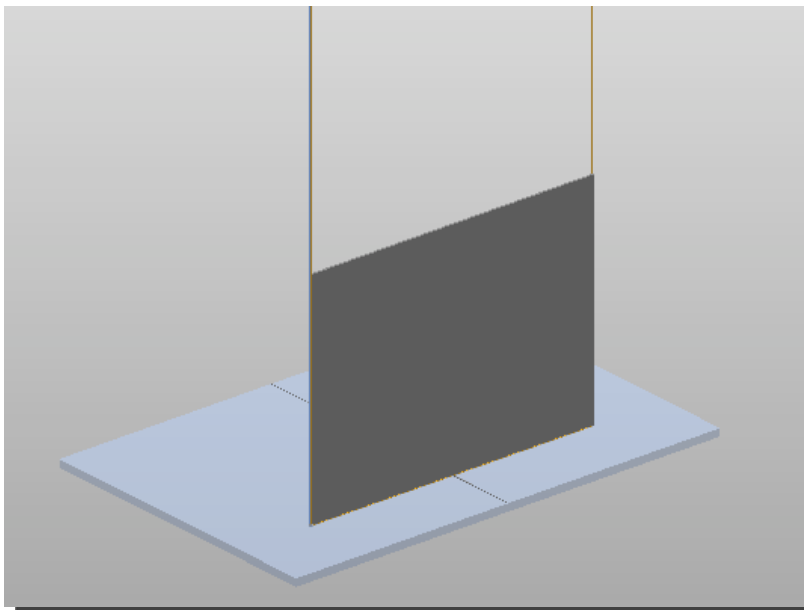


Figure 202

Making a Turntable Model

The default model style for every motor in the show is a winch, which moves linearly along a track. This works great for deck winches, lifts, and curtains which all move in a linear fashion.

However, a lot of shows include turntables. SpikeMark makes it easy to change a motor model from winch to turntable.

1. Select the motor you wish to adjust in the **Cue Grid**.
2. Scroll to the **Schematic** section of the **Motor Editing Pane** and select **Turntable** at the bottom of the screen (Figure 203).

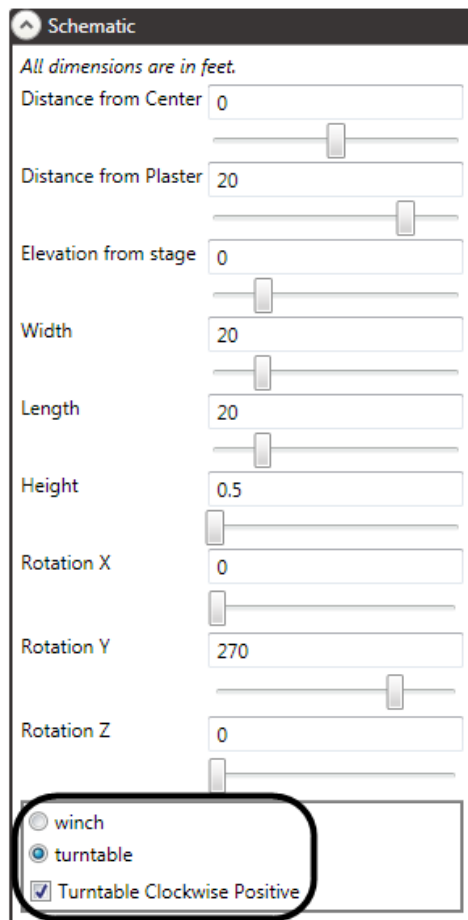


Figure 203

3. Use the **Distance from Plaster** and the **Width** sliders to position and resize the turntable (Figure 204 and Figure 205).

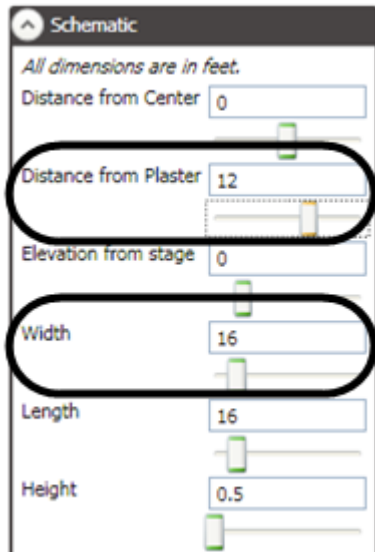


Figure 204

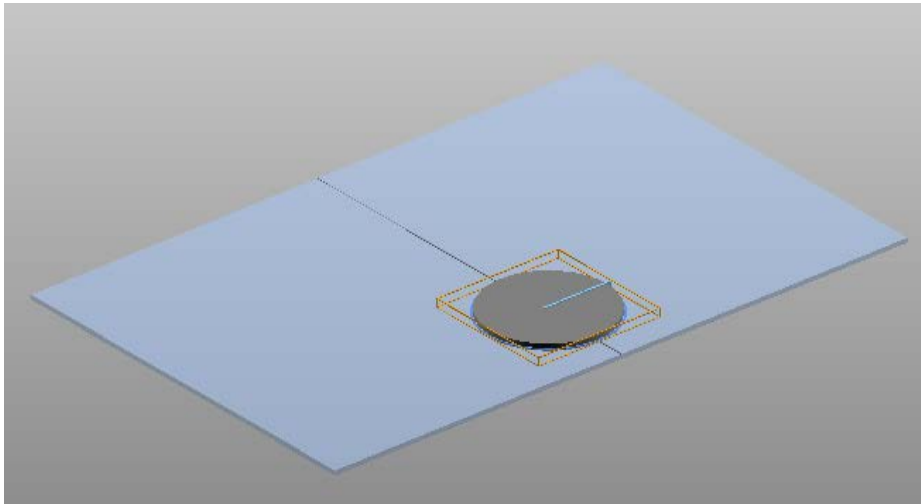
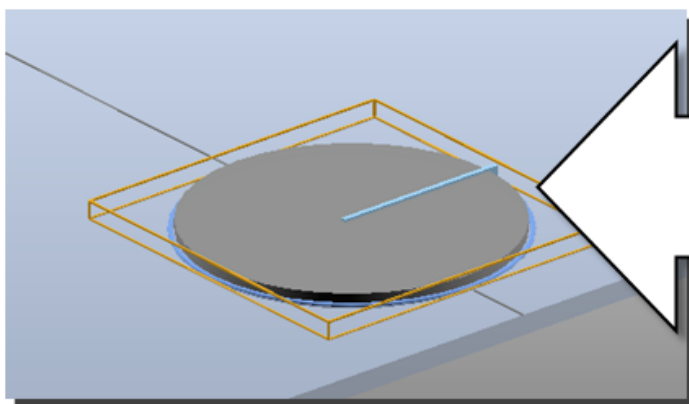


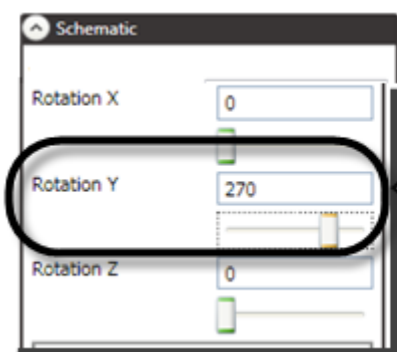
Figure 205

NOTE: By default, the turntable model orients position zero (“0 degrees”) towards Stage Left. You can change this orientation by changing the **Rotation Y** parameter (see Figure 206, Figure 207 and Figure 208).



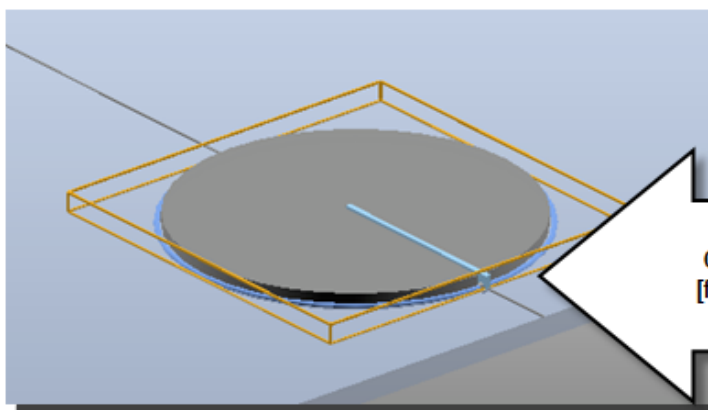
By default, the Turntable model orients position zero (0 degrees) toward Stage Left. You can move position zero by changing the **Rotation Y** parameter.

Figure 206



Use the **Rotation Y** parameter to spin position zero to wherever you need it on stage. *In this example, position zero will be at 270 degrees, or Down Stage.*

Figure 207



Changing the Rotation Y parameter to "270" [for example], puts position zero Down Stage.

Figure 208

Spikes

The **Spikes** feature is designed to mimic the usefulness of real spike tape. Rather than trying to remember bizarre motor position numbers (like 251.13”), you can now assign names to commonly used positions (‘on stage’ = 251.13”).

Often during the rehearsal process, the **Spike** positions need to be tweaked. If you were physically pushing the wagons around, you would just move the spike tape on the stage floor to a new position. Similarly, with the new **Spikes** feature, you simply edit the position of the **Spike** in one place and all cues that contain a motor that uses the **Spike** are automatically updated, saving you from manually having to make the change throughout all the cues where that motor is used.

Adding Spikes within Cues—Using the Spikes Position Button

1. Click on the applicable Cue in the **Cue Grid** that contains the motors you wish to set the Spikes for (Figure 209).

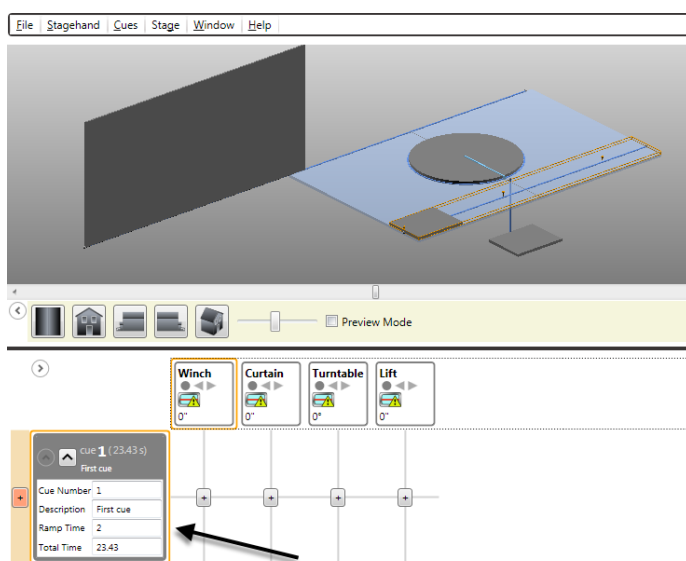


Figure 209

2. Select the motor to which you wish to add a **Spike** (Figure 210).

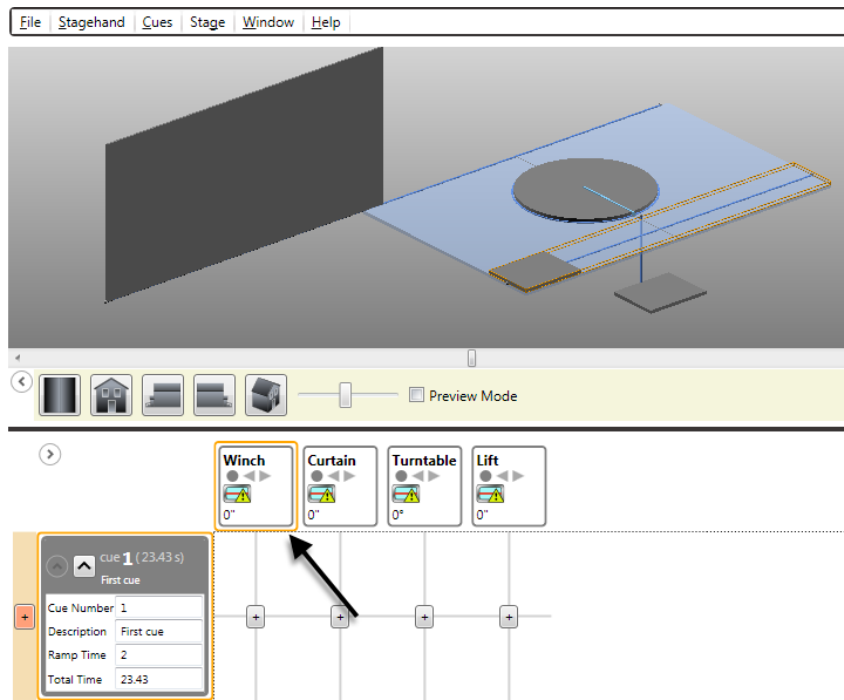


Figure 210

3. If you don't already have a movement for the selected motor in this cue, click the **Add Movement** button in the **Cue Grid** (Figure 211).

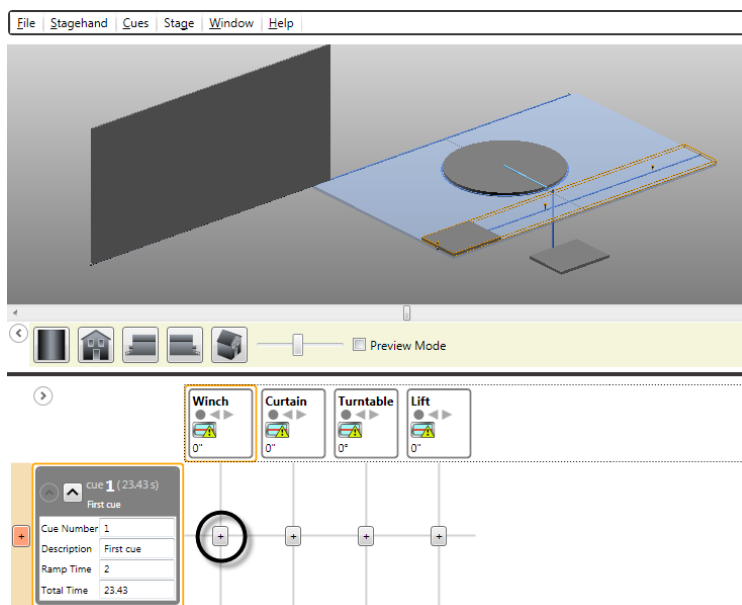


Figure 211

4. Click on the **Spike Position** button in the motor's movement window (Figure 212).

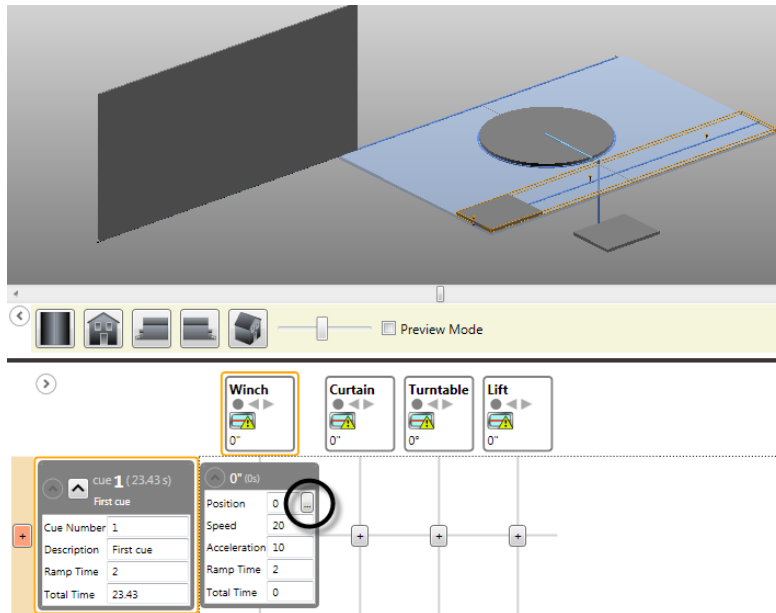


Figure 212

5. Click on the **Add Spike**  button in the Spikes window (see Figure below).

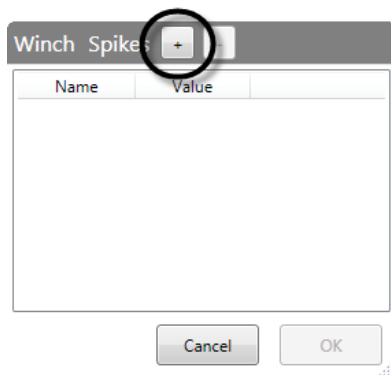


Figure 213

6. A new Spike will be added with the generic **Name** of “spike” and a **Value** of “0” (Figure 214).

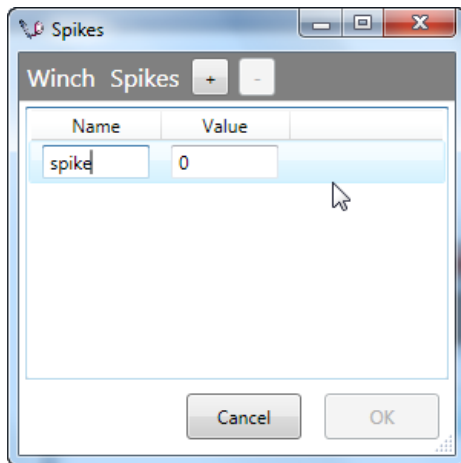


Figure 214

7. Enter a meaningful **Name** and the Spike position **Value**. Select the Spike and click **OK** (Figure 215).

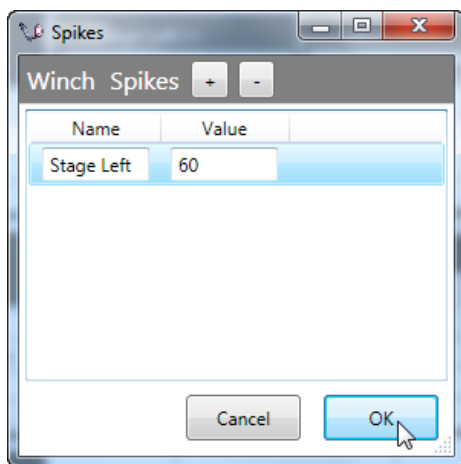


Figure 215

8. The selected Spike will display in the motor's movement for the cue (Figure 216). *In the example below, the Winch will move to the defined Spike—"Stage Left 60"—in Cue 1.*

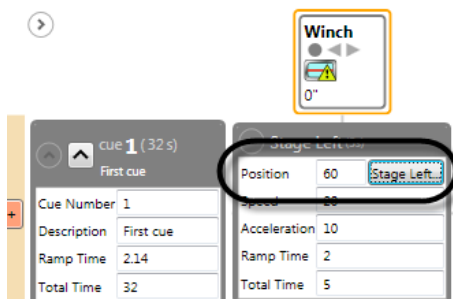


Figure 216

Adding Spikes within Cues—Using the Motor Editing Pane

1. In the **Cue Grid**, select the motor to which you wish to add Spikes (Figure 217).

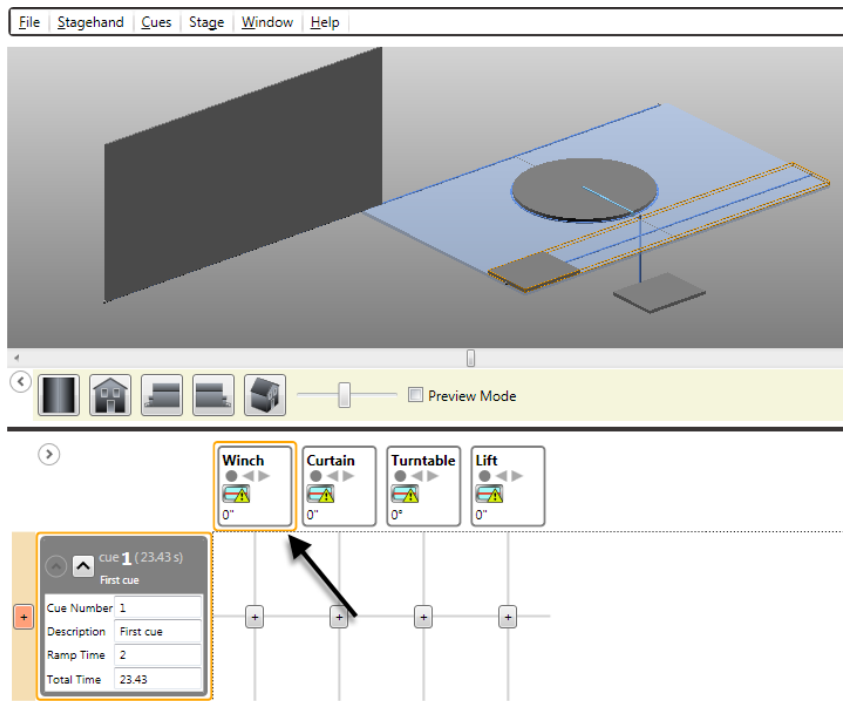


Figure 217

2. Scroll to the **Spikes** section in the **Motor Editing Pane** and click **Add Spike** (Figure 218).

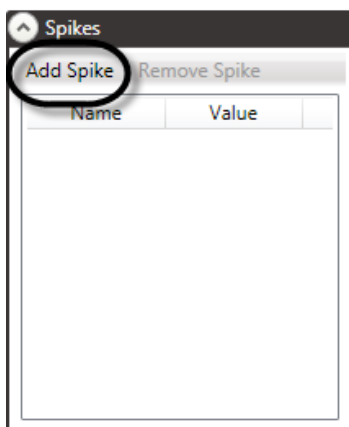


Figure 218

3. Enter a logical **Name** and the **Value** (the measurement where the Spike should be located).

*In the example in Figure 219, the **Name** of the Spike is “Stage Right” and the **Value**, or where the Spike will be placed, is position “0.”*

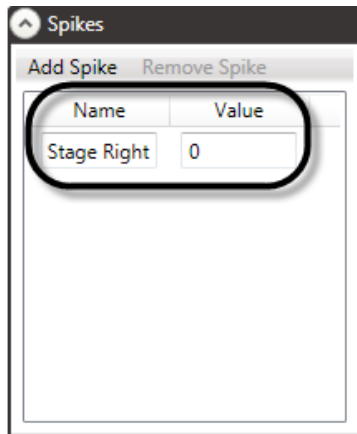


Figure 219

4. You can repeat steps 2 and 3 until you have all the Spikes for this motor.
In Figure 220, the Spikes “Stage Left” and “Center” have also been added.

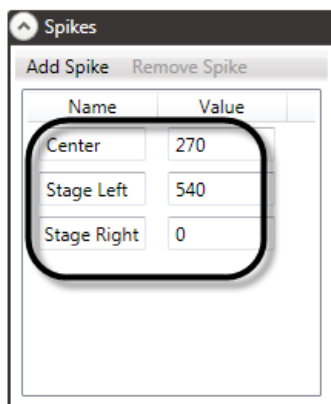


Figure 220

5. You can now set the position for the motors within any given cue by clicking on the applicable Cue in the **Cue Grid** (Figure 221).

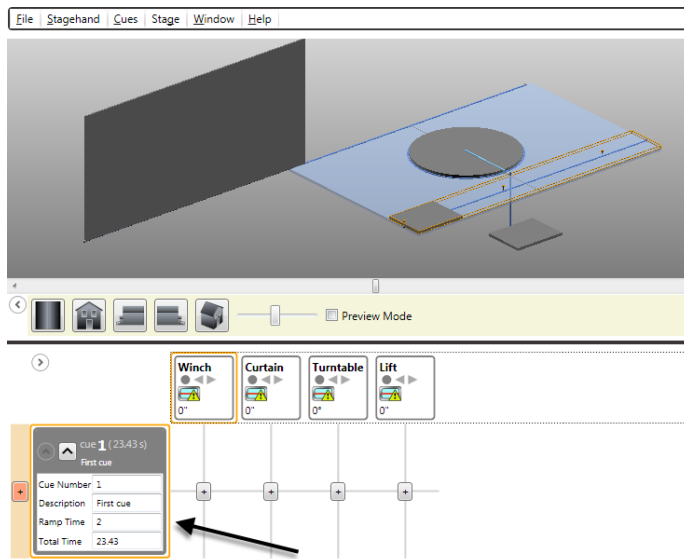


Figure 221

6. If you don't already have a movement for the selected motor in this cue, click the **Add Movement** button in the **Cue Grid** (Figure 222).

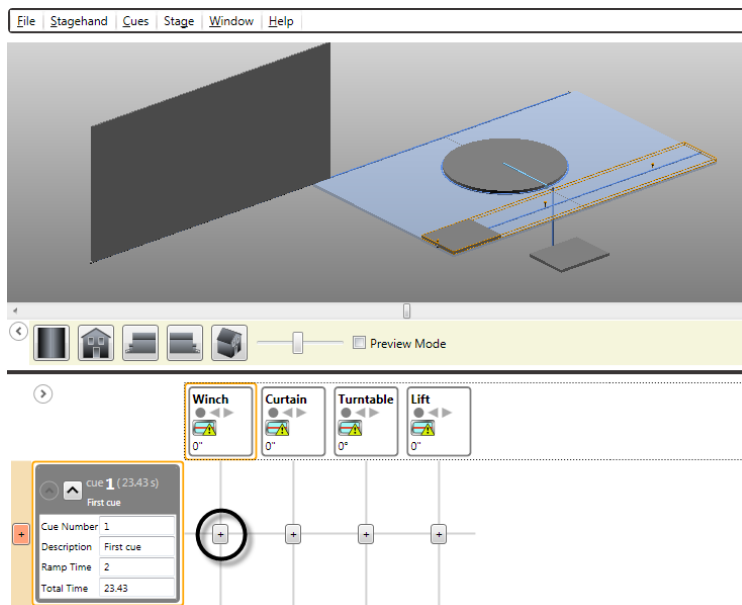


Figure 222

7. Click on the **Spike Position** button in the motor's movement window (Figure 223).

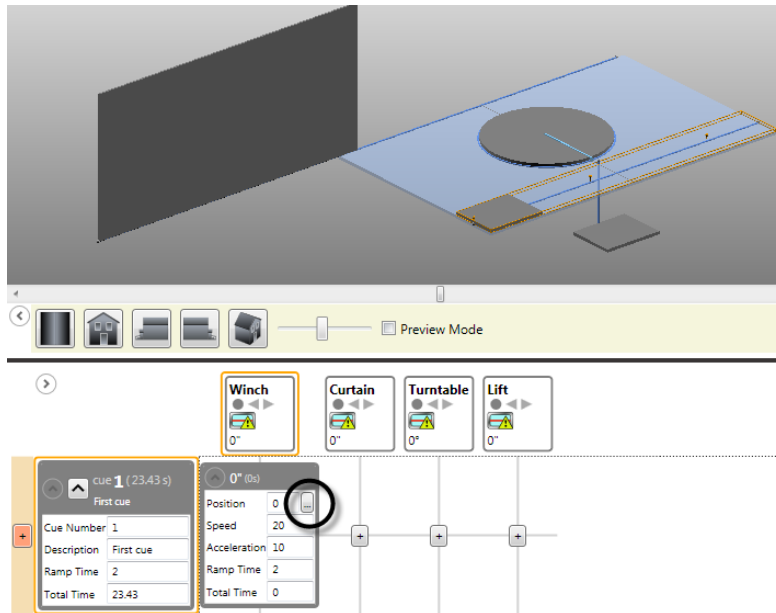


Figure 223

8. Select the Spike where you want this motor to move in this cue and click **OK**.
In Figure 224, the Spike “Center 270” has been selected.

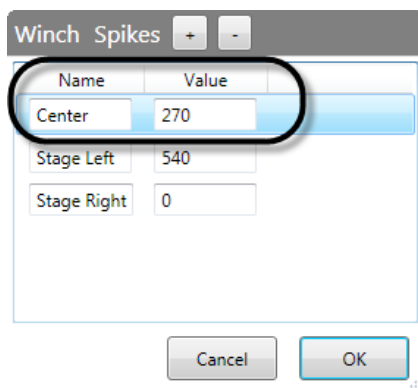


Figure 224

9. The selected Spike will display in the motor’s movement for the cue (Figure 225).
In the example below, the Winch will move to the defined Spike—Center, 270—in Cue 1.

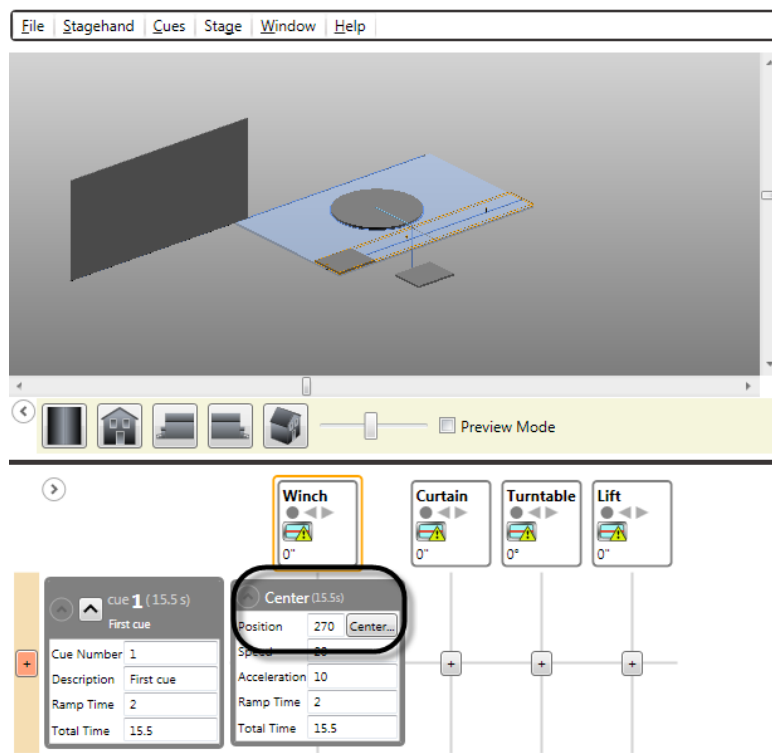


Figure 225

Changing a Spike Position

A key advantage to using **Spikes** is that you can edit the spike value and all cues that reference the spike will be updated. This makes it very easy to tweak position values and avoid tracking the changes manually through the entire show. For example, we've defined a spike named "midstage." However, during the rehearsal process it becomes evident that the "midstage" position needs to move onstage a quarter of an inch. We've already used this spike in a bunch of cues, so to update the spike and have the new position updated in all cues we just edit the spike.

1. In the **Cue Grid**, select the motor that has the Spike you wish to edit (Figure 226).



Figure 226

2. Go to the **Spikes** section of the **Motor Editing Pane** (Figure 227).

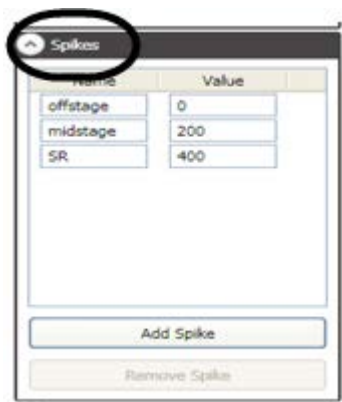


Figure 227

3. Edit the **Value** for the applicable **Spike**, and press the **TAB** key.
In the example below, the “midstage” Spike **Value** has been changed to “200.25” (Figure 228).

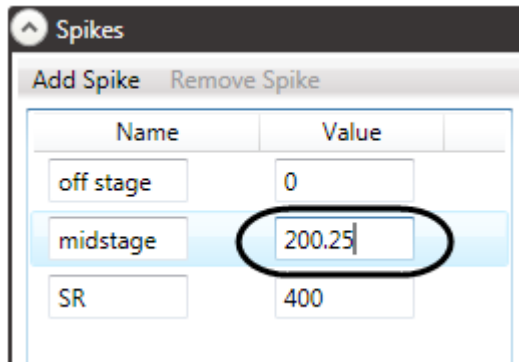


Figure 228

- The position values will update in the cues that reference the Spike you edited (Figure 229).

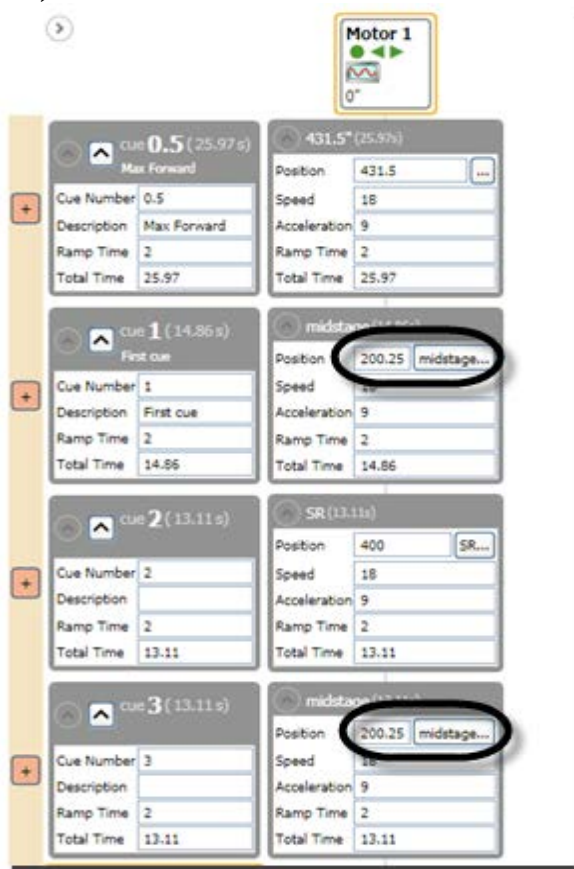


Figure 229

Selecting Spikes Using the Stage Model Viewer

- Click on the applicable Cue in the **Cue Grid** (Figure 230).

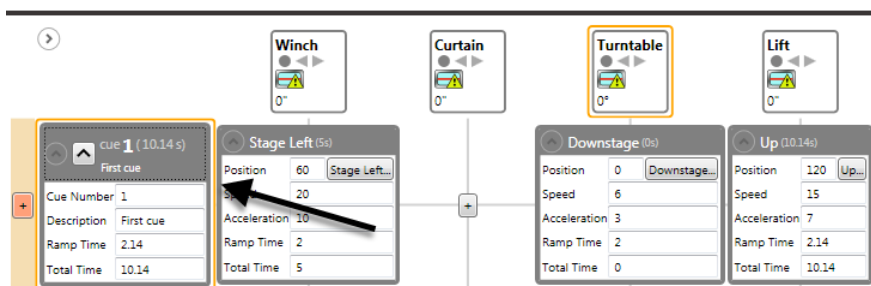


Figure 230

2. Check the **Preview Mode** box on the **Stage Model Viewer**. The motors that are included in the selected Cue will turn orange (Figure 231).

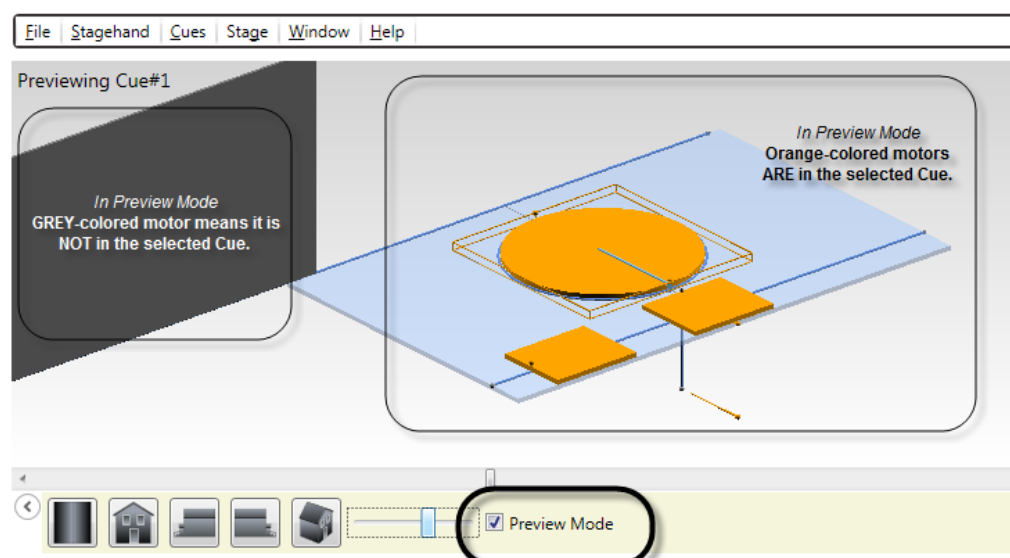


Figure 231

3. Select the motor to which you wish to add a Spike (Figure 232).

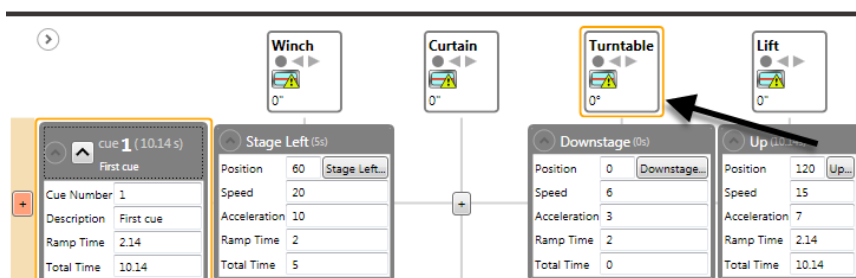


Figure 232

4. Zoom in and pan around the **Stage Model Viewer** to locate the Spike you wish to move the motor toward.
In Figure 233, the "Upstage" Spike has been selected.

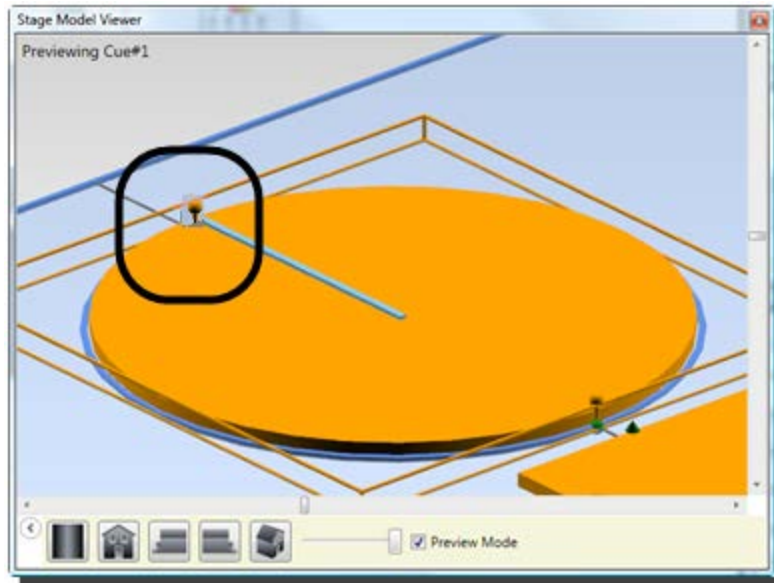


Figure 233

5. Note that the **Cue Grid** has updated to show the selected Spike (Figure 234).

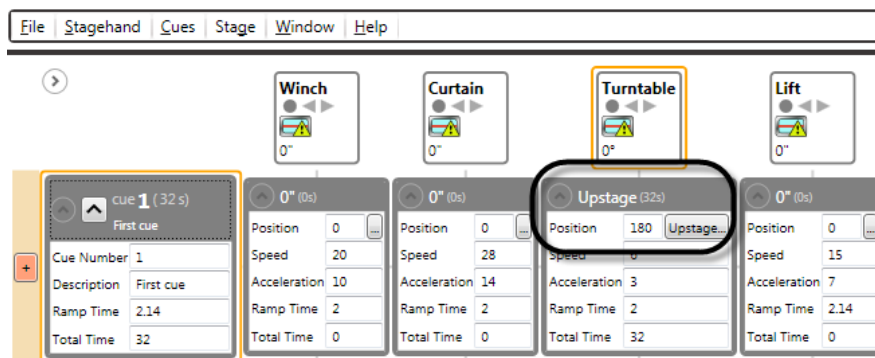


Figure 234

NOTE: When you select the **Preview Mode** field for a Cue, the **Stage Model Viewer** displays all the motors in GREY at the position of their last cue movement. This gives you an easy way to visualize where all the motors in the show will be at any given point in the show, regardless of whether the motor is moving in the previewed cue.

Running Shows

All of the setup, configuring, and cue writing is, in the end, just a prerequisite for what you really need to do—run a performance. As the balance of writing in this manual may indicate, running a show is actually the easiest part of the process. Thankfully, by doing all your preparation attentively, running shows is easy. We'll look at a few ways to execute cues, navigate through the show, and explain what to do when something doesn't work as you expected.

Creating a New Show

When SpikeMark first opens, it adds a single motor and a single cue to the show. You can start editing the motor and cue as a starting point for your show. If you would like to begin with a blank slate, you can create a new show by clicking **New** in the **File** menu (Figure 235).

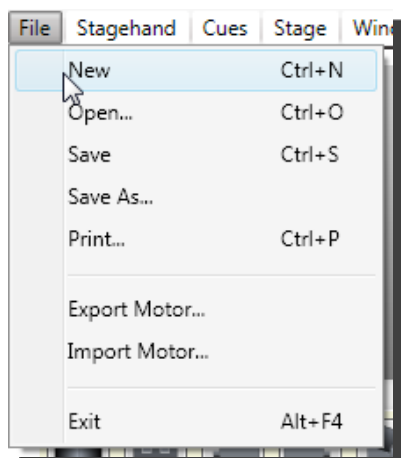


Figure 235

Setting up the Workspace

Once you've finished writing and editing cues, there are a few things you may want to arrange differently in the interface to see the appropriate amount of information. The way in which you like to set up the user interface is a personal preference, so the points outlined here are merely suggestions.

In the steps below, we'll expand the **Cue Grid** to take up the whole screen by double clicking on the splitter bars to minimize the **Motor Editing Pane** and the **Stage Model Viewer**. Next we'll add a **Stage Model Viewer** that will sit on top of the SpikeMark main screen by clicking **Stage Model Viewer** in the **Window** menu. Resize and arrange to your liking.

1. Expand the **Cue Grid** to its full size by double clicking on the **Stage Model Viewer** and the **Motor Editing Pane**'s Splitter Panes (Figure 236).

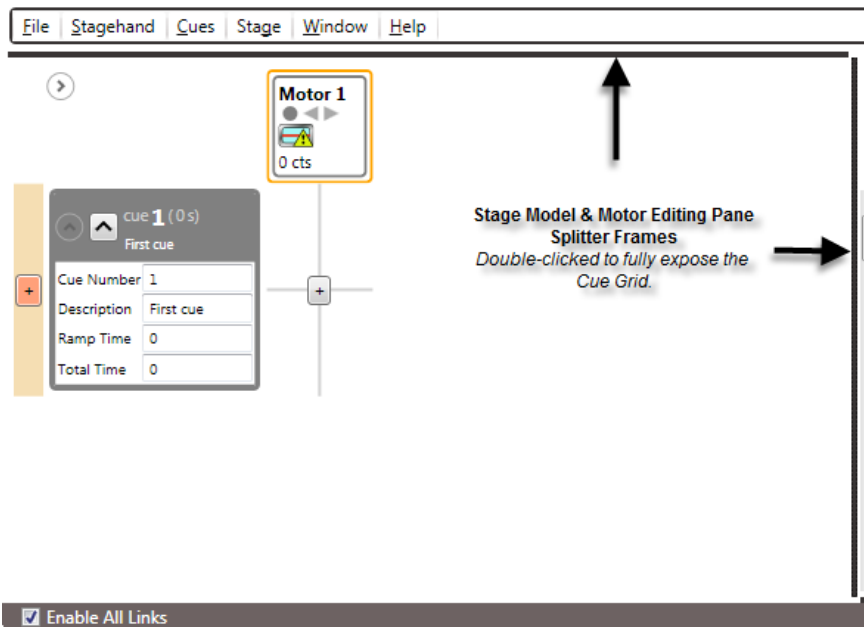


Figure 236

2. Hide all the speed and acceleration information for the cues by right-clicking in the **Cue Grid** and selecting **Collapse All** (Figure 237).
This will compact the Cue Grid while allowing you to see the time each cue will take and the positions of every motor (Figure 238).

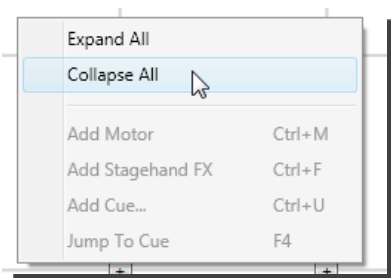


Figure 237

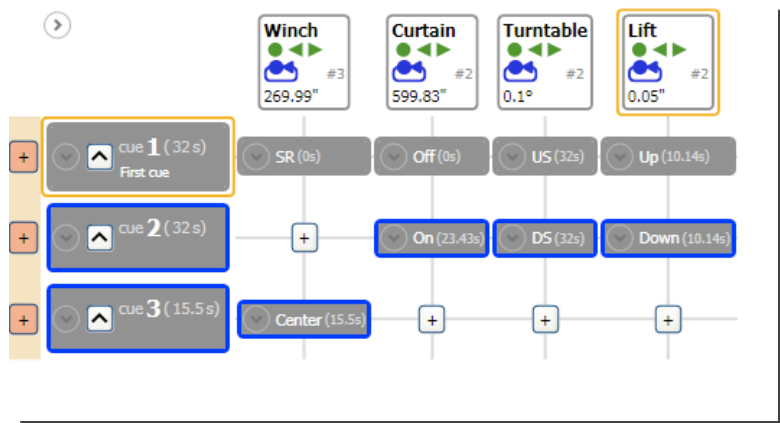


Figure 238

3. In **Window** menu, click **Stage Model Viewer** to create a floating **Stage Model Viewer**. Position the **Stage Model Viewer** in some unoccupied space of the **Cue Grid** (Figure 239) or drag it over to a second monitor.

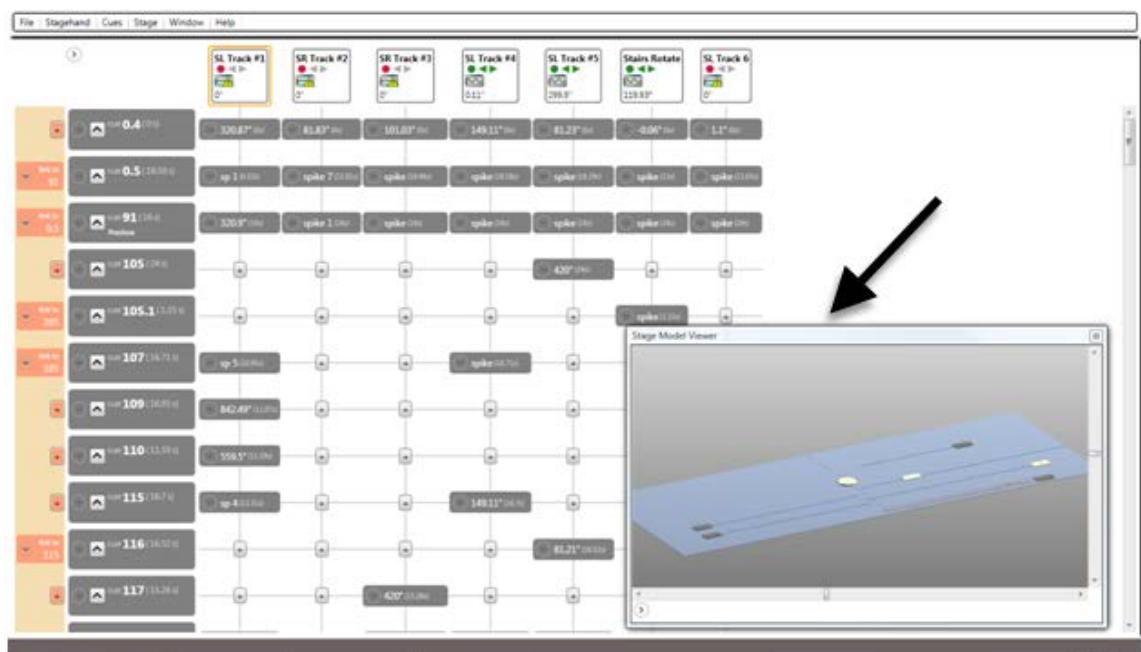


Figure 239

Rearranging the Cue Grid

By default, motors appear as columns in the **Cue Grid** in the order in which they were created. While that's sensible to SpikeMark, it probably bears no significance to you. Luckily you can rearrange the columns of the **Cue Grid** to follow the logic of your show.

1. Right-click in a blank section of the **Cue Grid** and select **Rearrange Stagehand Order...** (Figure 240).

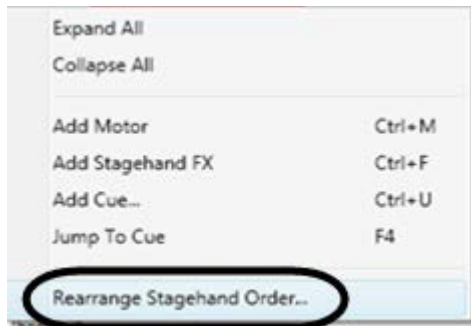


Figure 240

2. In the **Stagehand Display Order Editor** dialog box, select the motor that you wish to move and then use the up/down arrow buttons to rearrange the display order (Figure 241).

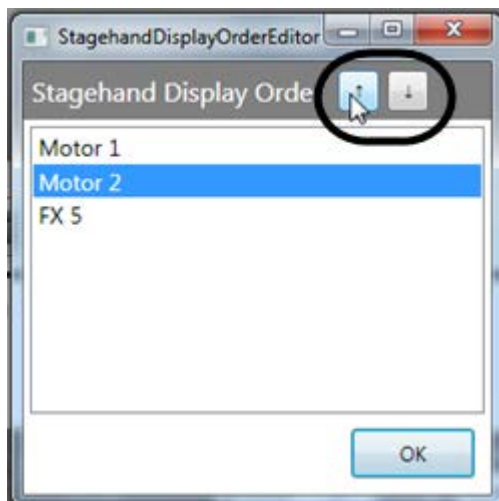


Figure 241

Loading a Cue

Executing a cue is a two step process. The first step is to load the cue. Loading the cue sends all the cue information out to the Stagehand Motor or FX and gets the motors ready to move. Typically, you'll load a cue when the stage manager gives you a "warning" for the cue.

NOTE: Cues that are **linked** do not need to be explicitly loaded, as the link will *automatically* load and execute its child cue.

There are a few ways to load a cue. The difference is a matter of personal preference.

1. Click the **Load Cue** button in the **Cue Grid** (Figure 242).

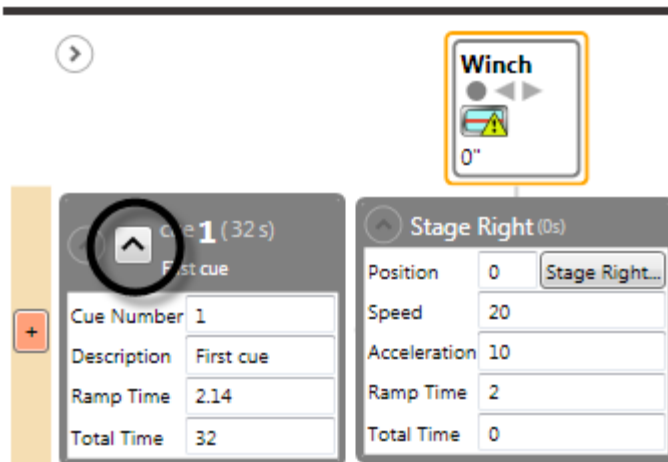


Figure 242

Or, click the **Jump To Cue** button on the **Cue Controller** in the **Motor Editing Pane** (Figure 243) and then select a cue from the Jump to Cue dialog box (Figure 244).
You can also type a cue number or description to jump to the cue.

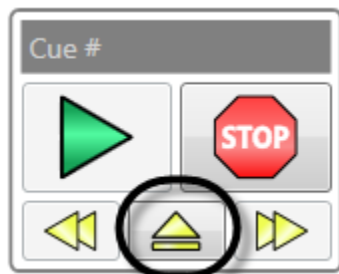


Figure 243

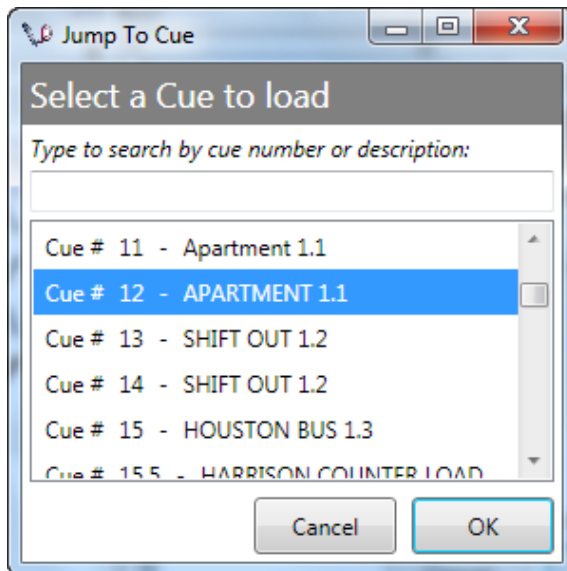


Figure 244

Or, if a cue was loaded or run previously, you can use the **Next Cue** and **Previous Cue** buttons in the **Cue Controller** (Figure 245).

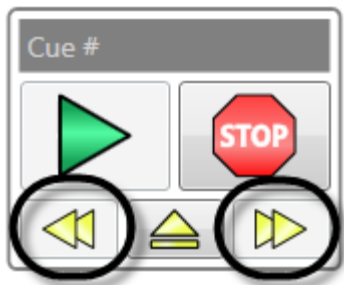


Figure 245

Or, press the **Load Cue** button on your Showstopper and select a cue from **Jump to Cue** dialog box.

Visual Indicators of Primed and Ready Movements

1. In the **Cue Grid**, the borders around the ready cue and ready movements change to red and the Stagehand column headers display a rocket ship countdown icon (Figure 246). *For a complete list of status icons and their meanings, see the “Motor Status” section on page 43 and the “FX Status” section on page 70.*

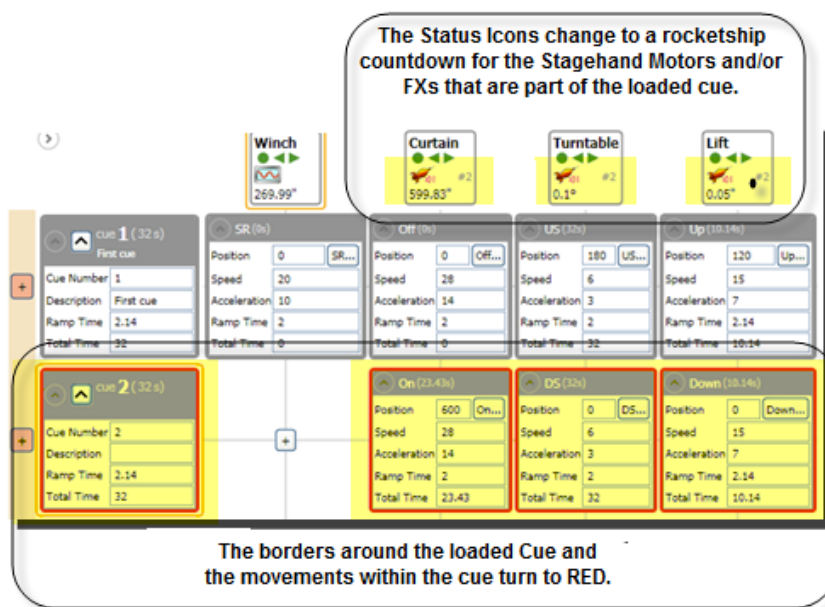


Figure 246

NOTE: If only some of the Stagehands were able to load the cue, the cue outline will turn pink indicating a partially loaded cue. This can occur if a Stagehand is not currently connected to SpikeMark.

- The status bar text updates in the bottom left corner of SpikeMark (Figure 247).



Figure 247

- The motor(s) that will move in the loaded Cue change to red and their target position(s) are shown in semi-transparent red inside the **Stage Model Viewer** (Figure 248). *Note that the Turntable's target position in the example below is the thin, semi-transparent red line that sits down stage, nearest the lift.*

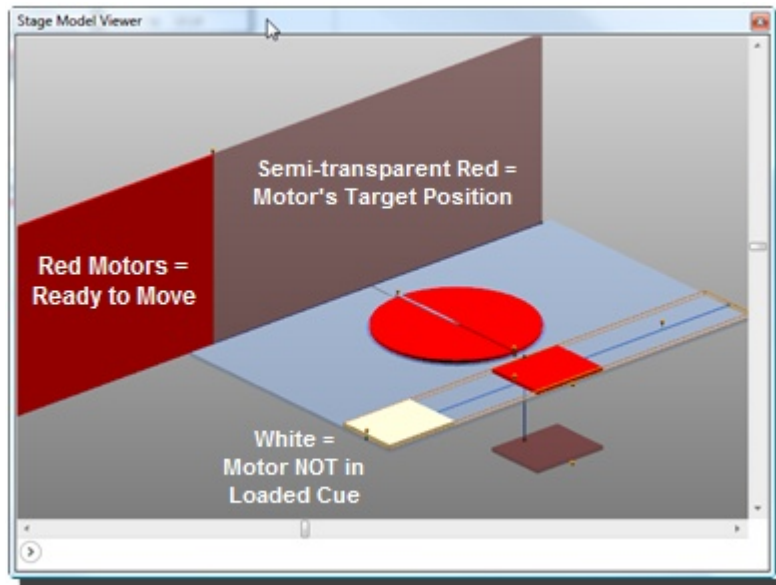


Figure 248

Running a Cue

Once a cue is loaded, you can start all the motors moving by running the cue. There are a few ways to run a cue.

1. You can press the **GO** button in the **Cue Controller** in the **Motor Editing Pane** (Figure 249).

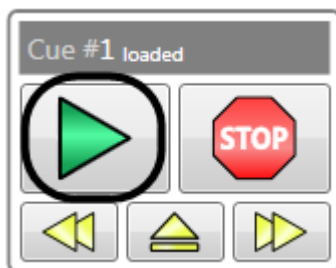


Figure 249

Or, click **Run Cue** in the **Cues** menu (Figure 250).

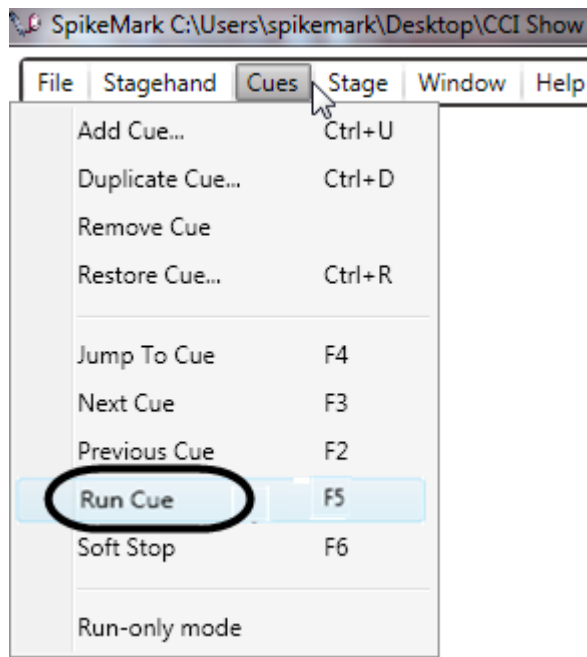


Figure 250

Or, press **F5** on the keyboard.

Or, press the **Go Cue** button on your Showstopper.

What Happens While a Cue Is Running

Once the cue is running, the interface changes to inform you exactly *what* is running.

1. In the **Cue Grid**, the border around the Cue that is running and motor movements within the running Cue change to green and the Stagehand status icons change (Figure 251). *For a complete list of status icons and their meanings, see the “Motor Status” section on page 43 and the “FX Status” section on page 70.*

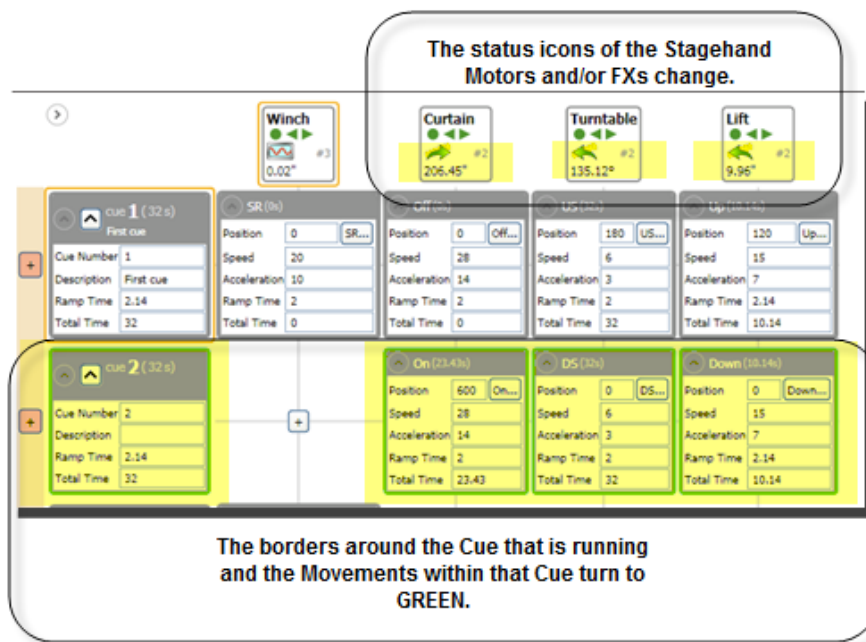


Figure 251

2. The motor(s) that are moving in the Cue change to green inside the **Stage Model Viewer** (Figure 252).

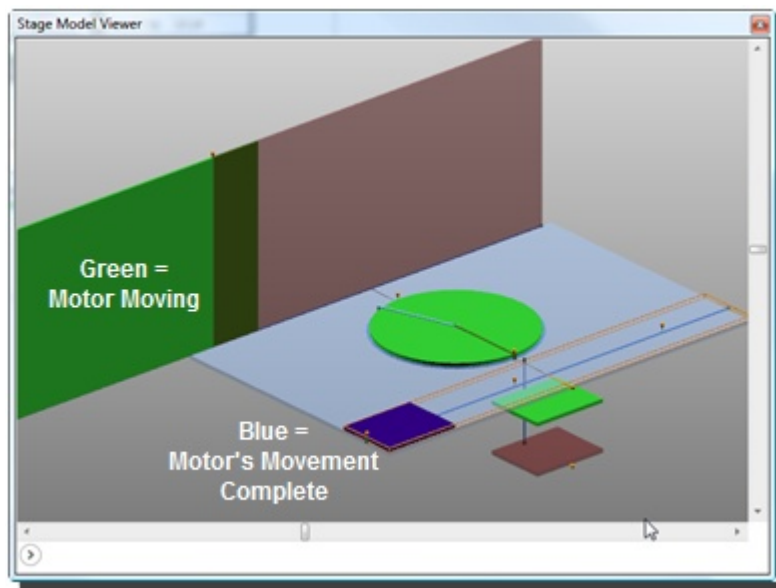


Figure 252

3. The status text in the **Cue Controller** displays which cue is running (Figure 253).



Figure 253

NOTE: If you have *multiple cues* running at the same time, the Cue Controller only shows the last cue started.

4. The **Status Bar** text in the bottom left corner of the SpikeMark screen displays which cue is running (Figure 254).

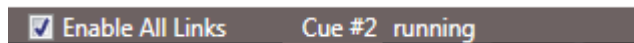


Figure 254

NOTE: If you have *multiple cues* running at the same time, the Status Bar only shows the last cue started.

What Happens as Motor Movements and Cue(s) Complete

1. As the motor movements and the cue complete, the interface shifts colors again and the borders in the **Cue Grid** turn blue (Figure 255).

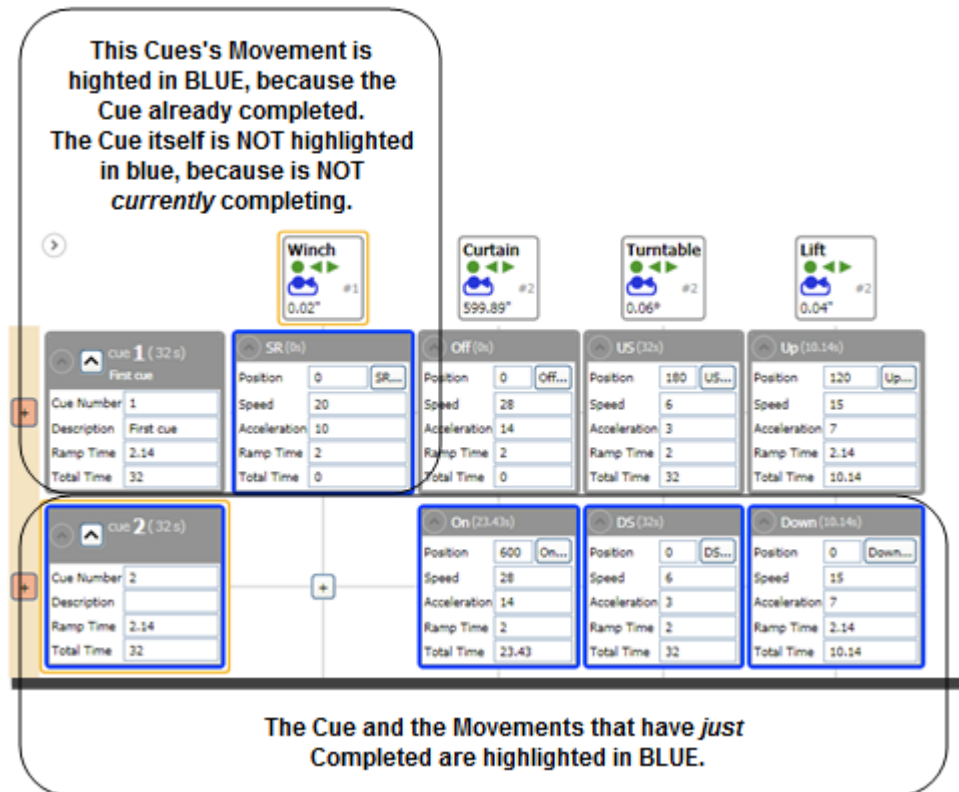


Figure 255

- The motor(s) that have completed their movements in the Cue change to blue inside the **Stage Model Viewer** (Figure 256).

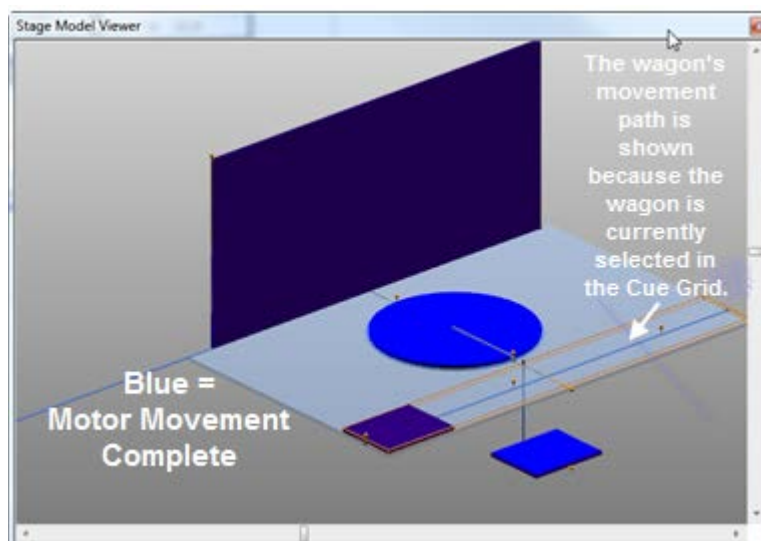


Figure 256

- The **Cue Controller** in the **Motor Editing Pane** on your screen updates its status text (Figure 257).

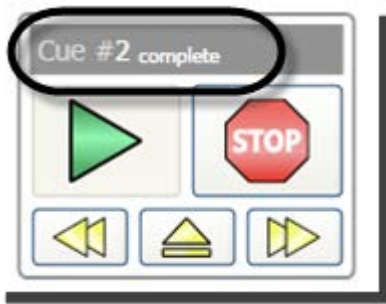


Figure 257

4. The **Status Bar** text in the bottom left part of the SpikeMark screen is updated (Figure 258).



Figure 258

Deciphering Stagehand Motor Status During a Show

In the heat of a performance, you are likely to feel compelled to obsessively check that your motors and FXs are all connected, happy, and ready to run. There's no shame in that compulsion, it's what makes you a good automation operator. If there is a problem, you want to know about it so that you can fix it quickly and be ready to run your next cue. SpikeMark feeds that hunger for status reports with a few key signals in the interface.

The **Cue Grid** header cells give you quick status information about every Stagehand Motor and FX. Let's take a look at a Motor header first (Figure 259).

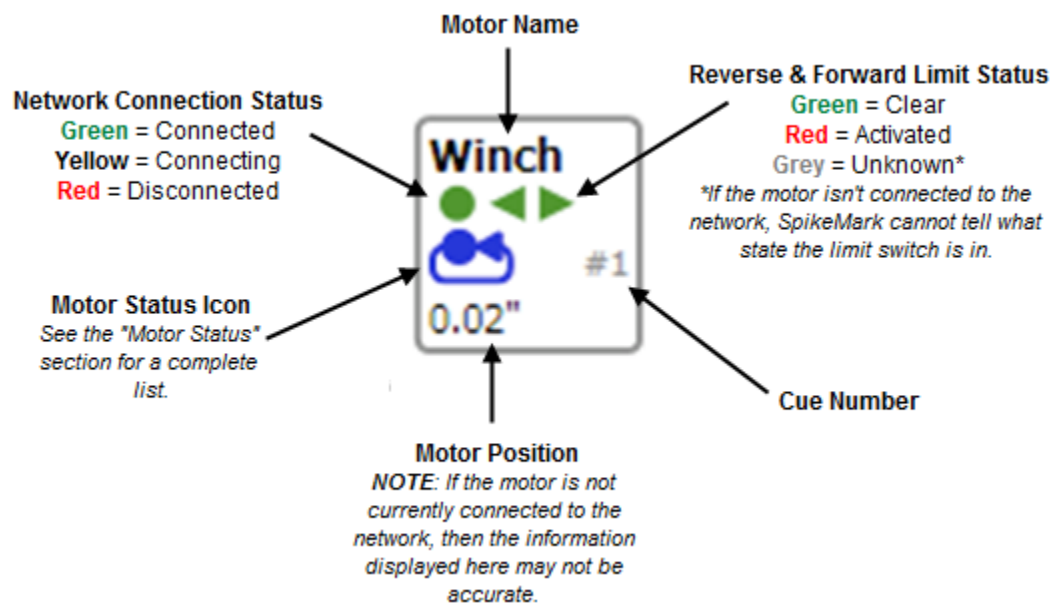


Figure 259

If the motor has been Deactivated (by deselecting the **Active** box in the **Cue Behavior** section of the **Motor Editing Pane**), the background of the motor and all its movements will turn grey (Figure 260).

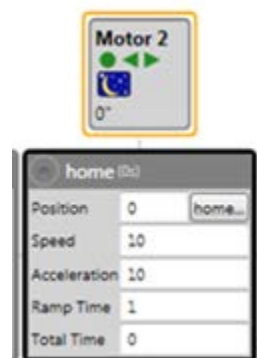


Figure 260

The **Cue Grid** header cells give you quick status information about every Stagehand FX (Figure 261).

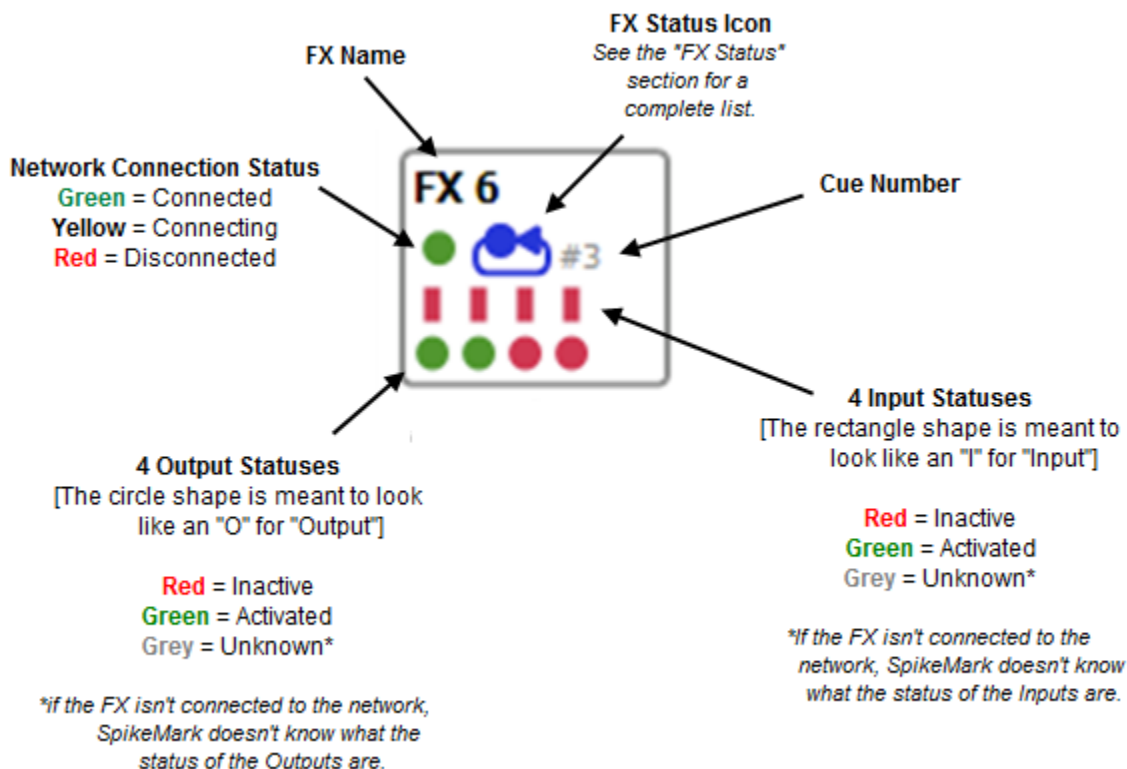


Figure 261

If the Stagehand FX has been Deactivated (by deselecting the **Active** box in the **Cue Behavior** section of the **FX Editing Pane**), the background of the motor and all its movements will turn grey (Figure 262).

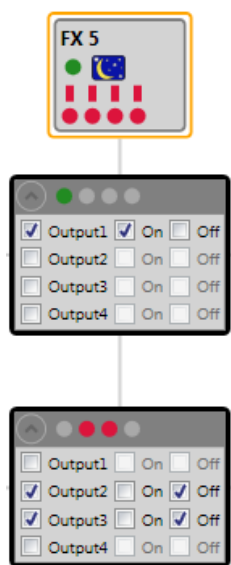


Figure 262

Restoring a Cue

SpikeMark's **Cue Restore** feature allows you to move backwards through a show during rehearsal. **Cue Restore** will put every motor back into its proper position for a specific cue, regardless of whether the motor was moving in that cue or had its last movement in a previous cue. If that sounds confusing, it is straightforward in practice. Let's take a look at how the feature works and you'll see that it does exactly what you need when jumping backwards in rehearsal.

In the following example, assume that we just ran through Cue #4 and the stage manager needs to re-run Cue #4 because Jeff, the lighting designer, needs to look at the transition again. As you glance at the cue sequence below, you can see that we need to reset the stage so that we are sitting in Cue #3. But there's a problem. You can't just run Cue #3 to reset. If you were to just run Cue #3, then Motor 1 would be left in its "home" position. So you would need to run Cue #3 and Cue #2. Not a big deal in this scenario, but as you have more motors and more complex cues, restoring gets to be a real pain. Thankfully, **Cue Restore** can handle this easily.

1. Right-click on Cue #3 and select **Restore** (Figure 263).

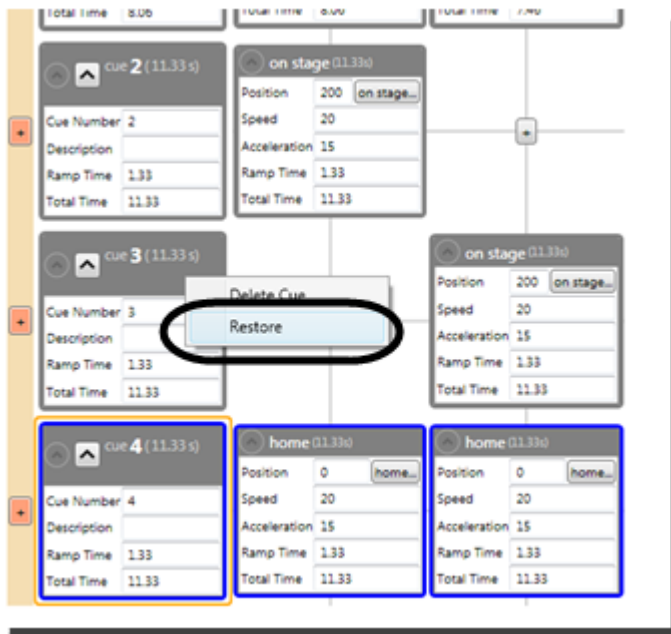


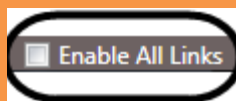
Figure 263

2. A list is presented showing that both **Motor 1** and **Motor 2** need to move in order to restore Cue #3 (Figure 264).



Figure 264

NOTE: The checkbox in the status bar at the bottom of the SpikeMark main screen—"Enable All Links"—can be used to turn off cue links. The **Cue Restore** feature *automatically* deselects the **Enable All Links** field [pictured below] in the status bar at the bottom of the SpikeMark main screen, so that cue links are not accidentally triggered during a Cue Restore.



- To start moving motors back into their correct positions, select a single movement from the list and then press the **Load Movement** button in the **Restore Cue** window (Figure 265). Load and run the movements one by one.

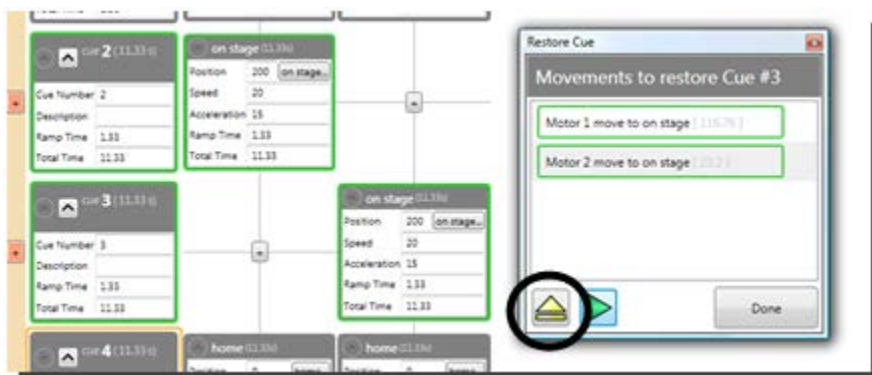


Figure 265

NOTE: You don't have to wait for each movement to complete before starting the next movement. But you must confirm that restoring a movement *will not cause a collision on stage*.

4. After one movement is loaded, you can press the **Run** button in the **Restore Cue** window (Figure 266). Then—if it is safe—load the next movement and run it as well. *In the example below, two movements are running concurrently by starting Motor 2 before Motor 1 completes.*

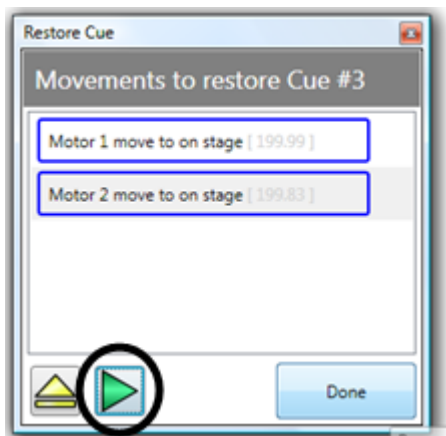


Figure 266

5. When you are finished restoring all the motors, press the **Done** button to close the **Restore Cue** window (Figure 267).



Figure 267

6. Click **OK** in the **Enable Cue Links** dialog box to re-enable all cue links (Figure 268). *If the motors have NOT finished moving, click **Cancel**.*

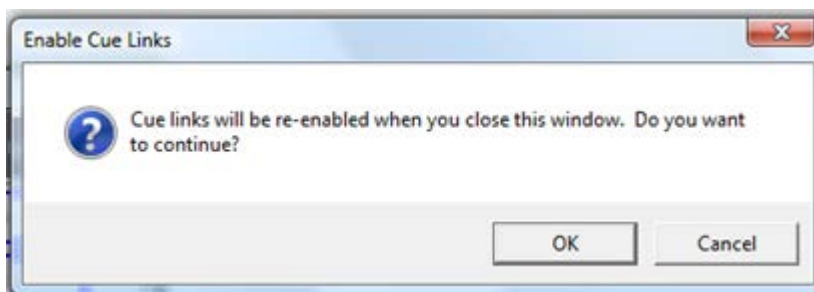


Figure 268

7. Notice that in the status bar in the bottom left-hand corner of the SpikeMark main screen, the **Enable All Links** box is once again selected (Figure 269).



Figure 269

8. Cue #4 is restored and ready to run again. Repeat the steps to restore the cue again, if needed.

Troubleshooting

When automation works well, it's a fun and wondrous technology. When automation doesn't work it's hair-pulling, frustrating black-magic. If you've made it to this part of the manual, I assume you are at the hair-pulling stage. I'll try to help answer some of the more common problems and solutions here. If these suggestions don't address your problem, give us a call, send an email, or use the discussion forums.

Contact Us

Phone 401-862-2980 (free tech support for the first 90 days, after that \$30 / half-hour fees may apply at our discretion).

Email support@creativeconners.com (always free, response times vary from minutes to a couple of hours).

Access our Frequently Asked Questions web page at <http://creativeconners.com/pages/faq.html>

Web forums can be found at <http://creativeconners.com/phpBB3/>

Issue	Possible Solutions
The Stagehand Network Connection Doesn't Work	The most common reason for this is that the IP address between the Stagehand and the computer are not compatible. See the "Setting Up a Network Connection" section on page 19.
	Check your network cabling. Swap out the cables, preferably with a cable you know is good from another working Stagehand.
	Try resetting the IP address of the Stagehand to a slightly different address, and then update the address in SpikeMark. If the network server in the Stagehand has stalled, resetting its IP address will get it started again.
	Make sure that you have a network hub between the computer and the Stagehand. If you are trying to connect directly from the computer to the Stagehand you need a special type of network cable—Crossover Cable.
	Swap out the network hub.

Issue	Possible Solutions
A Motor Didn't Start To Move When I Ran A Cue	Check to see if the motor is sitting on a limit switch. Limit switch status is shown in the Cue Grid and the Motor Editing Pane .
	Make sure you have a Proportional Gain of at least 1 in the Tuning section of the Motor Editing Pane .
	Make sure that the cue had a speed and acceleration more than zero (0).
	Check the activity log to get some information about what happened. In the Window menu, click Log Viewer . The Log Viewer is displayed and it shows all the actions that SpikeMark is executing behind the scenes. You can sort the log by clicking on the column headers.
SpikeMark Showed A Strange Error Message Then I Had To Quit	That's a bug, and we need to fix it. Please email a copy of your show file, a copy of the Activity Log, and a brief description of any pertinent information to support@creativeconners.com .

Issue	Possible Solutions
I Can't Find The Log File To Email It To Support	From the Help menu, click Reveal Log File. The Application Data folder is hidden by default on some Windows installations. To view the folder, click the Start Menu and select Computer. Press the Alt key to reveal the File menu. Click Tools, Folder Options..., View and then click Show hidden files and folders. Log files are stored in: C:\Users\spikemark\AppData\Roaming\Creative Conners, Inc\SpikeMark\2.0.0.0
A Motor Starts To Run In A Cue But Stops Abruptly Before Getting To Spike	The motor isn't tuned well enough to reach its target position: 1. Tune the motor better if possible. See the "Motor Tuning" section on page 51. 2. In the Cue Behavior section of the Motor Editing Pane, deselect the Strict Timing Mode field. This will let the motor keep trying to get to target position even if the tuning is sub-optimal. 3. If the motor is showing a Position Error fault and you are certain that nothing is physically stuck in the machinery, you can raise the Max Position Error value in the Position section of the Motor Editing Pane. 4. If the motor is showing a Position Error fault, you can turn off the Abort On Position Error option inside the Position section of the Motor Editing Pane. <u>This is not the ideal thing to do</u>. If the encoder becomes broken or disconnected, nothing will stop the motor from running away. Since no feedback is being sent to the Stagehand, the Stagehand will send increasing amounts of power to the motor until it is at full speed and you will need to use the Emergency Stop to shutdown the motor. See the "Setting the Max Position Error" section on page 57 for more information.
The Motor Runs Full Speed In The Wrong Direction	The motor and encoder polarity are out of sync. Swap two of the power leads in the motor plug or swap the encoder signals A & /A with B & /B inside the encoder plug.
SpikeMark Not Responding/ Running Slowly	Change the motor's refresh status rate. See the "Changing the Status Refresh Rate" section on page 62 for instructions.

Appendix A - Basic Motion Control Concepts

From the perspective of an audience member, motion control is magic. From the technician's perspective it should not be magic. The combination of Stagehand motor controllers and SpikeMark software create a motion control system that everyone can use, but it is still important to know some basic concepts to avoid mistakes and help troubleshoot your system.

Power

Depending on the model of Stagehand that you are using, you need to supply either 120vac or 230vac power to each Stagehand. The Stagehand will convert this power into something suitable for your motor. The output voltage, or frequency for AC motors, is varied to achieve variable motor speed.

The Stagehand tracks the motor's position and then relays that position to SpikeMark to display on-screen. If a Stagehand loses power, the next time it powers up it will reset its position to zero (0). This is very important, because if a motor is sitting 10' from centerline and the power is interrupted, when the power comes back on the Stagehand will think that the motor is now at 0' from centerline. In this case, the motor needs to be driven back to the true 0' spike (usually a piece of spike tape or limit switch) and reset to 0', either through SpikeMark (using the **Reset Position** button in the **Position** section of the **Motor Editing Pane**) or by cycling power to the Stagehand.

The best operating procedure is to return all motors to zero (0) at the end of every performance. If power is lost overnight when no one is around, then it won't affect the Stagehand's position information. A pre-show check should be done every night to insure that every motor is truly at the correct position.

Since all position information is stored in the Stagehand, the PC running SpikeMark can be shut down at any time without affecting position information whatsoever.

Position Feedback

How does the Stagehand know how to change speeds? Good question. There are two factors that affect the speed of a motor. The first is the target speed that is programmed for each cue. In SpikeMark you tell each motor how fast to move in each cue. This information is sent to the appropriate Stagehand, which in turn supplies some voltage to the motor and watches to see what happens. If the motor moves faster than the Stagehand expected, it reduces the voltage. If the motor moves slower than expected, the voltage is raised. This cycle continues at a rapid pace until the motor reaches the desired position.

In order for the Stagehand to know how far and how fast the motor is moving in relation to the voltage it is supplying, it needs an encoder. An encoder is a simple device that creates electronic pulses as it spins. The encoder is physically connected to the motor so that as the motor spins the encoder spins. If the encoder were disconnected from the motor, the Stagehand would not see any movement regardless of whether the motor was actually rotating or not. This could cause the motor to race, because the Stagehand would keep supplying more and more voltage to the motor hoping to see an increase in speed. A similar situation would occur if the polarity of the motor was reversed from the encoder.

There are safety features that you can set up to prevent a motor from racing, such as **Max Position Error**, which tells a Stagehand to shutdown if it doesn't get good response from the motor. This feature is discussed in the "Setting the Max Position Error" section on page 57.

Another safety precaution to guard against motor racing is discussed in the "Over-Travel Limits" section below.

Motor Tuning

The cycle of watching encoder pulses and adjusting motor voltage is controlled by the tuning parameters of the Stagehand. These tuning parameters are often referred to as the PID filter, which stands for Proportional, Integral, and Derivative gains. You enter numeric values for this filter into SpikeMark, which will then send the filter parameters to a Stagehand. While the mathematics of this filter is somewhat complex, the practical application is not. We walked through motor tuning in greater detail in the "Motor Tuning" section on page 51.

Over-Travel Limits

If the encoder position is wrong because of loss of power, physical disconnect from the motor, or some equipment failure, a physical limit switch can be used to protect the motor from running too far. These limit switches, when struck, will prevent the motor from moving any farther in the detected direction (either forward or reverse). The Stagehand will only permit a movement in the opposite direction until the limit switch is cleared.

These basic principles of motion control will become more concrete and intuitive as you become more familiar with the operation of SpikeMark and Stagehand.

Index

- 3-Dimensional Stage View, 15
- Abort On Position Error, 57
- Acceleration, 31, 86, 89, 90
- Active Mode
 - controlling with an FX, 62
 - cue links and, 60
 - for an FX, 72
 - safety concerns for a motor and, 60
 - safety concerns for an FX and, 72
 - setting for a motor, 60
 - setting for an FX, 71
- Add Link Button, 97, 99, 100, 101
- Add Movement Button, 126, 131
- Add Spike Button, 127
- Adding
 - a new show, 138
 - cues, 83
 - FX movements to a cue, 91
 - motor movement to a cue, 84
 - motors, 40
 - new FX, 65
- Advanced
 - section of Cue Grid, 63
- Auto-repair, 78, 80
- Child Cue, 96, 97, 99, 100, 101
 - illustrated version of, 97
- Completed Cue
 - indicators of, 35
 - what happens after, 148
- Completion Link, 98
 - creating, 97
 - illustrated version of, 97
 - when not to use, 105, 106
- Configuring
 - FX, 67
 - Motors, 40
- Confirming Encoder Feedback, 25
- Connecting
 - to a motor, 79
 - to a motor or FX, 23
 - to an FX, 79
- Creative Connors
 - contact information, 157
 - online tutorials website, 13
 - technical support email address, 157
 - web forums, 157
- Cue Behavior
 - for a motor, 60
 - for an FX, 71
- Cue Completion Mode
 - strict timing, 58
 - target tolerance, 58
- Cue Controller, 35, 37, 145, 147, 148, 149
- Cue Description, 83, 84
- Cue Grid, 37, 41
 - rearranging, 140
- Cue Links, 96–106
 - child cues, 96
 - cue completion, 97
 - deleting, 102
 - infinite loops, 97, 105
 - motor position, 100
 - multi-speed/multi-position, 103
 - parent cues, 96
 - time link, 99
 - using FX inputs, 101
- Cue Number, 83, 84, 151, 152
- Cue Outline Color Meaning
 - blue, 149
 - green, 147
 - pink, 144
 - red, 144
- Cue Restore, 153
- Cue Restoring, 153
- Cues
 - adding, 83
 - adding a motor movement to, 84
 - adding an FX movement to, 91
 - changing how fast a motors moves within, 88
 - changing motor position in, 87
 - changing total time of, 90
 - completed, 148
 - deleting, 94
 - linking, 96
 - loading, 141

- pile on, 96
 - removing a motor's movement from, 93
 - restoring, 153
 - running, 34, 145
 - what happens while running, 146
 - writing, 32
- Curtain Model
 - making a, 118
- Deactivated
 - FX, 72, 152
 - motor, 61, 151
- Deactivating FX Input Action, 73
- Default Acceleration, 31
- Default Speed
 - setting, 29, 30
- Delay Field, 99
- Delete Movement, 93
- Derivative Gain, 52, 161
- Disconnect From All, 81
- Disconnecting
 - from a motor, 80
 - from an FX, 80
- Distance from Center, 114, 117, 120, 121
- Distance from Plaster, 114, 117, 120, 121, 122
- Elevation From Stage, 114, 117, 120
- Enable All Links, 154, 156
- Encoder Counts
 - decreasing, 26
 - increasing, 26
 - increasing vs. decreasing, important notes, 45
- Encoder Feedback
 - confirming, 25
- Error Message
 - in SpikeMark, 158
- Ethernet link
 - confirming that it works, 19
- Filter Loaded, 54
- Forward Limit Status, 151
- Forward Physical Limit, 50
- Frequently Asked Questions Website, 157
- Front View Button, 110
- FX, 65–76
 - activating outputs, 71
 - active mode, 71, 72
 - adding a new, 65
 - adding movement to a cue, 91
 - configuring, 67
 - connecting to, 79
 - connecting to, quick start, 23
 - deactivating action for, 73
 - disconnecting from, 80
 - Input Action, 73
 - inputs, 93
 - outputs, 93
 - removing input action for, 74
 - renaming, 68
 - renaming inputs and outputs, 68
 - soft stopping action for, 74
 - status icon descriptions, 70
 - status information in cue grid, 152
- FX Editing Pane, 38, 65
- FX Input Link
 - creating, 101
- FX Status, 70, 152
- FX Support, 16
- GO Button, 145
- Height, 117, 119, 121
- Icons
 - for FX status, 70
 - for motor status, 43
- Input Action
 - for an FX, 73
 - removing for an FX, 74
- Input Statuses, 152
- Inputs, 93
 - renaming for an FX, 68
- Installing SpikeMark, 8
 - Microsoft's .Net 3.5 framework, 12
 - registering software, 12
- Integral Gain, 53, 161
- Integral Limit, 54
- IP Address, 78
 - illustrated version of, 19
 - rules for setting, 19, 22
 - segments of, 19, 22
 - Setting for your computer, 19
 - unique for each stagehand and computer, 19
- Isometric View Button, 111
- Jump To Cue button, 142

- Left View Button, 110
- Length, 117, 119, 121
- Lift Model
 - making a, 115
- Limits
 - forward, 48
 - physical, 50
 - reverse, 48
 - rules for setting, 49
- Linking Cues, 96–106
- Load Cue Button, 34
- Load Movement Button, 154
- Loaded Cue Indicators, 34
- Loading a Cue, 141
- Log File, 159
- Lurching
 - what to do if motor is, 52
- Main Screen, 37
- Manual Controls, 25, 29, 44, 47, 48
- Manually Jogging a Motor on Screen, 44
- Max Forward Position, 29, 48
 - rules for setting, 49
- Max Position Error, 57
 - maximum allowable, 57
 - setting, 57
- Max Speed, 90
- Min Reverse Position, 28, 48, 114
 - rules for setting, 49
- Motion Control Concepts, 160–61
 - Motor Tuning, 161
 - Over-Travel Limits, 161
 - Position Feedback, 160
 - Power, 160
- Motor, 40–64
 - active mode, 60
 - adding a new, 40
 - adding movements to cues, 84
 - changing position of in a cue, 87
 - changing speed automatically, 104
 - changing speed manually, 103
 - configuring, 40
 - connecting to, 79
 - connecting to, quick start, 23
 - deactivated, 60, 151
 - disconnecting from, 80
 - doesn't start, 158
 - lurching, what to do if, 52
 - manually jogging with on screen controls, 44
 - movement parameters, 85
 - removing movement from a cue, 93
 - renaming, 42
 - resetting position value for, 50
 - runs full speed in wrong direction, 159
 - scaling position, 45
 - selection for a position link, 101
 - smooth out motion for, 53
 - speed in cue, 88
 - starts but stops abruptly, 159
 - tuning, 36, 56
 - what to do after motor is tuned, 57
- Motor Cue Completion Mode, 58
- Motor Editing Pane, 37, 41
 - using to add a spike, 129
- Motor Name, 151
- Motor Position, 151
 - changing using 'ping-pong' effect, 105
- Motor Status, 151
 - deciphering during a show, 151
 - icon descriptions, 43
- Motor Tuning, 51–56, 161
 - flow chart, 56
- Movements
 - visual indicators of primed and ready, 143
 - what happens when complete, 148
- Multi-position Cue Links, 103
- Multi-speed Cue Links, 103
- Network Connections, 77–81
 - auto-repair, 78, 80
 - breaking, 80
 - connecting, 77
 - disconnected, 77
 - errors, 77
 - not working, 157
 - setting up a, 19
 - status for a motor, 151
 - status for an FX, 152
- Network Settings
 - editing, 77
- Network Status, 24
- New Show
 - creating, 138

- Newest Features in SpikeMark
 - 3-dimensional stage view, 15
 - FX Support, 16
 - reset position, 16
 - spikes, 13
 - Stage Model Viewer, 15
 - streamlined interface, 14
 - target tolerance mode, 16
 - time-based cues, 14
- Next Cue Button, 143
- Nightly Motor Position Resetting, 50, 160
- Online Tutorials, 13
- Outputs, 71, 93
 - manually activating for an FX, 71
 - renaming for an FX, 68
 - statuses of, 152
- Over-Travel Limits, 161
- Parent Cue, 96, 98, 99, 100, 101
 - illustrated version of, 97
- Parts of Main Screen
 - Cue Controller, 37
 - Cue Grid, 37
 - FX Editing Pane, 38
 - Motor Editing Pane, 37
 - Stage Model Viewer, 37
- Physical Connections
 - illustrated example of, 18
 - setting up, 18
- Physical Limits, 50
 - when to set, 18
- PID Filter, 161
- Pile On Cues, 96
- Ping Pong
 - a motor's position, 105
- Plan View Button, 109
- Polling Interval, 62, 63
- Poorly Tuned
 - what to do when motor is, 56
- Position, 85, 87
 - changing in a cue, 87
- Position Feedback, 160
- Position Link, 105, 106
 - creating, 100
 - illustrated example of, 97
- Position Scale, 27, 46, 47
- Position Scaling Factor
 - setting, 26
- Position Units, 27, 47
- Power, 160
- Power Loss
 - what to do, 50
- Preset View Buttons, 109
- Preview Mode, 136, 137
- Previous Cue Button, 143
- Proportional Gain, 36, 52, 161
- Quick Start Guide, 18–36
- Ramp Time, 86, 87, 89, 90, 91
- Rearrange Stagehand Order, 140
- Rearranging the Cue Grid, 140
- Remove Cue, 94
- Renaming
 - a Motor, 42
 - an FX, 68
 - inputs and outputs for an FX, 68
- Reset Position, 16, 50
- Reset Position Button, 46
- Resetting a Motor to Arbitrary Position, 16
- Resetting Motor Position Each Night, 50
- Restoring a Cue, 153
- Reverse Limit Status, 151
- Reverse Motor Polarity, 45
- Reverse Physical Limit, 50
- Right View Button, 111
- Rotation X, 116
- Rotation Y, 114, 123
- Rotation Z, 116, 121
- Running Cues, 34, 145
 - indicators of, 35
 - what happens while, 146
- Running Shows, 138–56
 - deciphering motor status during, 151
 - restoring a cue when, 153
- Safety, 7
- Sampling, 53
- Scaling Factor
 - setting, 45
 - setting, quick start, 26
- Schematic, 112, 116, 118, 122
- scrollbars
 - using in the Stage Model Viewer, 109
- Setting
 - default speed, 29

- FX active mode, 71
- Max Position Error, 57
- Min Reverse and Max Forward positions, 48
- motor cue completion mode, 58
- motor's active mode, 60
- soft limits, 28
- stage dimensions, 111
- the Position Scaling Factor, 26
- Setting Up
 - IP address for your computer, 19
 - Network Connections, 19
 - Physical Connections, 18
 - SpikeMark workspace, 138
- Smooth Out a Motion, 53
- Soft Limits
 - setting, 28
- Soft Stopping
 - using an FX, 74
- Speed, 86, 88, 89, 90
 - allowing SpikeMark to update, 89
 - changing directly, 88
 - changing for a motor automatically, 104
 - changing for a motor manually, 103
 - changing for motor in a cue, 88
 - setting default, 29
- Spike Button, 87
- Spike Position Button, 126, 131
 - using to add a spike, 125
- SpikeMark
 - already installed, 8
 - error message, 158
 - frequently asked questions website, 157
 - is not responding, 159
 - main screen, 23, 37
 - Newest Features, 13
 - online tutorials, 13
 - technical support email address, 157
 - web forums, 157
- Spikes, 13, 82, 125–37
 - adding using spike position button, 125
 - adding using the Motor Editing Pane, 129
 - changing position of, 133
 - selecting via the Stage Model Viewer, 135
- Splitter Panes, 39, 139
- Stage Dimensions
 - setting, 111
- Stage Model Viewer, 15, 37, 107–24
 - as pre-tech tool, 107
 - floating, 140
 - making a curtain model in, 118
 - making a lift model in, 115
 - making a turntable model in, 121
 - making a winch model in, 112
 - navigation of, 108
 - opening in a separate window, 108
 - scrollbars, 109
 - setting the stage dimensions in, 111
 - using to select a spike, 135
- Stagehand Loses Power
 - what to do, 50
- Status
 - FX, 70
 - motor, 43
- Status Bar, 144, 148, 150
- Status Information
 - for a motor in the cue grid, 151
 - for an FX in the cue grid, 152
- Status Refresh Rate, 62
 - changing the, 62
- Streamlined Interface, 14
- Strict Timing Mode, 58
- System Requirements, 8
- Target Tolerance Mode, 16, 58
- Time
 - changing for cues, 90
- Time Link, 99, 105, 106
 - creating, 99
 - illustrated version of, 97
- Time-Based Cues, 14
- Total Time, 86, 87, 89, 91
- Troubleshooting, 157–59
- Tuned
 - what to do after motor is, 57
 - what to do when motor is poorly, 56
- Tuning
 - basic for a motor, 36
 - motors, 51–56
- Turntable Model
 - making a, 121
- Vertical Flying Curtain, 121

- Visual Indicators of Primed and Ready Movements, 143
- Web Forums, 157
- Website
 - for frequently asked questions, 157
 - for online tutorials, 13
- Width, 117, 119, 122
- Winch Model
 - making a, 112
- Workspace
 - setup, 138
- Writing Cues, 32, 82–95
- Zoom Slider, 108